Extração de Tabelas em Imagens Digitais por meio de HTML5 e JavaScript

Nayana Andrade Campelo
Escola Superior de Tecnologia, EST
Universidade do Estado do Amazonas, UEA
Manaus, Brasil
nac.eng@uea.edu.br

Neide Ferreira Alves
Escola Normal Superior, ENS
Universidade do Estado do Amazonas
Manaus, Brasil
nfalves@uea.edu.br

***Resumo*—Este trabalho descreve o desenvolvimento de uma aplicação de processamento de imagem, usando HTML5 e JavaScript, para identificar e extrair tabela em arquivo de imagem digital. Uma vez recebida a imagem, a aplicação, por meio de algoritmos como a Transformada de Hough e o Detector de Harris, busca identificar a estrutura da tabela e, posteriormente, gera uma nova imagem somente com a tabela identificada.**

***Keywords*-Reconhecimento de Tabelas; Extração de Dados; Transformada de Hough.**

***Abstract*—This paper describes the development of an image processing application using HTML5 and JavaScript, to identify and extract table into digital image file. Upon receipt of the image, the application through algorithms such as Hough Transform and Harris Detector, search identify the table structure and generate a new image only with identified table.**

***Keywords*-Tables Recognition; Extracting Data; Hough Transform.**

I. Introdução

A mobilidade permitida por equipamentos como *tablets* e *smartphones* leva o usuário a buscar aplicações em rede, como a Internet. Inúmeras aplicações para *desktop* ou mesmo os pacotes de escritório estão migrando para páginas da *web*. O desenvolvimento de aplicações de Processamento Digital de Imagens (PDI) na web permite que o usuário aplique suas técnicas a qualquer momento, bastando ter acesso a rede, porém é necessária uma linguagem que permita o desenvolvimento e mesmo a manipulação sem comprometer o fluxo de dados na rede. A linguagem *HTML5* juntamente com *JavaScript* dá esse suporte, tanto para equipamentos móveis quanto para os computadores *desktops*.

A migração de aplicações de PDI para *web* torna-se essencial para o usuário acessar e aplicar a qualquer momento, de acordo com suas necessidades. Entre estas aplicações, há a identificação de tabelas, assunto abordado neste trabalho.

Conforme [1] a função primordial do processamento digital de imagens é a de fornecer ferramentas para facilitar a identificação e a extração de informação contida nas imagens, para posterior interpretação.

Partindo deste conceito, de análise e manipulação das imagens, pode-se produzir outras imagens, estas contendo apenas as informações almejadas, de igual modo, extrair a tabela organizada estruturalmente, sintaticamente e semanticamente.

Este artigo apresenta a estratégia utilizada para a identificação e extração da tabela em arquivos de imagem. A Transformada de Hough e o detector de Harris foram utilizados. Todo o processo é apresentado ao longo deste trabalho, que é organizado da seguinte forma: a primeira seção consta de uma breve descrição do que se trata o presente documento, a segunda seção apresenta trabalhos relacionados na literatura; a terceira seção detalha a estratégia para a identificação e extração de tabelas; a quarta seção apresenta a ferramenta desenvolvida, a qual dá apoio a proposta deste trabalho, além de alguns testes de desempenho realizados; finalmente, a quinta seção apresenta as conclusões e desenha linhas para trabalhos futuros e a sexta e última seção lista as referências citadas ao longo do texto.

II. Trabalhos relacionados

Vários pesquisadores propuseram formas de identificação e extração de tabelas. Esta seção descreve alguns trabalhos e também ferramentas que estão próxima da proposta desta pesquisa.

Segundo [2], a dificuldade encontrada na avaliação e reconhecimento de estrutura de tabelas deve-se, principalmente, ao fato de não existir normas e/ou padrões nos documentos tomados como base, consequentemente, a comparação por Algoritmos de Reconhecimento de Estrutura de Tabela se torna inviável, acentuado pela coerência humana que os analisa.

Estratégias de avaliação, regras de classificação e modelagem são dispostas com a finalidade de eliminar ambiguidades dos resultados gerados pelas ferramentas, algoritmos, métodos e técnicas que envolvem a extração, manipulação de estrutura e conteúdo de tabelas [2].

Porém, outro fator negativo, devido a falta de padronização de documentos, também é visto na avaliação de desempenho de ferramentas populares de geração de documentos e/ou layout de tabelas. Um apontamento considerável, é o não compartilhamento de informações entre as ferramentas de edição, todas as referências de formatação/edição da tabela criada/editada numa dada ferramenta são perdidas quando usadas em outra ferramenta, não facilitando em nada o trabalho de edição de tabelas que por si só é difícil, demorado e propenso a erros [3].

Novos modelos, algoritmos e métodos tem sido estudados e propostos a fim de solucionar este dilema, no entanto ainda permanecem limitados quanto ao tamanho das tabelas, ou seja, apresentaram dificuldade em resolver o problema de disposição da tabela quando há um grande número de linhas, colunas e configurações de células, outra dependêcia também se mostra em relação aos recursos de hardware [4].

É mencionado em diversos artigos e de conhecimento em várias frentes de pesquisas a problemática – "Minimização de tabela para que caiba numa dada largura de página" e "Encontrar a tabela com altura mínima para uma dada largura de página" – automatizar este processo tem sido objeto de estudo e trabalho nas últimas décadas [4]. Este problema não é de fácil resolução, especialmente quando a tabela contém texto.

Tem se buscado também a extração dos dados (conteúdo da tabela) de forma que a disposição destes dados leve a uma interpretação consistente, permitindo resultados consideráveis e reconhecíveis. Baseados assim, em *Notação Wang* (Xinxin Wang propôs uma "tabela" de tipo abstrato de dados onde cada dimensão lógica é definida por uma árvore de categorias de domínios rotulados), bem como a *Ferramenta de Abstração de Tabelas* (TAT – *Table Abstraction Tool*. Uma tabela TAT Admissível, se TAT puder extrair sua correta notação Wang) [5].

No entanto, algumas pesquisas tiveram um bom andamento em suas propostas e resultados, como o "pdf2table: Um método para extrair informações de tabela em arquivos PDF" [6] e "Extraindo informações tabulares de arquivos de texto" [7].

A proposta [6] é desenvolvida sobe várias heurísticas que juntas reconhecem e decompõem tabelas de arquivos PDF e armazenam os dados extraídos em formato de dados estruturado (XML) para uma facilitar a reutilização. Um protótipo, também foi implementado, o que dá ao usuário a capacidade de fazer os ajustes nos dados extraídos, como editar o conteúdo da célula ou inserir linhas e colunas. O trabalho conseguiu bons resultados, porém apresenta limitações, uma delas é devido à complexidade da tarefa e as heurísticas utilizadas, que não pode cobrir todas as possíveis estruturas de tabelas, não se pode assumir que a abordagem sempre retorna resultados corretos. Outra limitação da ferramenta é que ela é baseada nos resultados da ferramenta *pdftohtml*. Se esta ferramenta retorna informações erradas ou nenhuma informação, a abordagem não pode ser aplicada, logo esta é considerada como a limitação principal, porque o usuário nada pode fazer e a interface gráfica também não vai ajudar.

A proposta [7] trabalha na localização e extração de tabelas e seus conteúdos a partir de imagens de documentos. O algoritmo consiste de quatro fases. Em primeiro lugar, utilizando técnicas de processamento de imagem, a imagem do documento é segmentada e analisada para isolar potenciais áreas de tabela. Em segundo lugar, essas áreas são passadas para um motor de OCR, que produz a saída de texto. Enquanto a maioria das pesquisas na área de análise e reconhecimento de tabela tem focado na análise da imagem formada de *pixels*, a abordagem se baseia nos avanços em *software* de reconhecimento óptico de caracteres (OCR) que é capaz de preservar a geometria da tabela, inserindo linhas em branco e caracteres de espaço para indicar separação semântica significativa no texto da tabela. Na terceira fase, o texto é analisado para isolar o início e o fim da(s) tabela(s). Finalmente, a análise local é realizada para isolar os componentes da tabela (blocos de título, células e rodapés).

Vale ressaltar que na pesquisa [7] foram selecionadas tabelas de dois domínios distintos, artigos de revistas científicas e relatórios financeiros. Os resultados foram satis-fatórios, as tabelas técnicas de artigos de revistas científicas foram mais fáceis de localizar e extrair do que as tabelas financeiras. Isto é devido ao método relativamente uniforme de apresentar dados técnicos. Embora existam normas para relatórios financeiros, o layout é substancialmente mais variado e mais difícil de detectar com os algoritmos atuais. A saída do motor de OCR, muitas vezes produz espaçamento imperfeito e caracteres incorretos. Daí a necessidade de dar mais ênfase à análise semântica de tabelas através de exame de conteúdo, bem como melhores modelos topológicos para descrever as tabelas, para corrigir esses erros. Além disso, na fase de processamento de imagem, seria útil incluir mais informações para a fase de análise de tabela, tais como localização de linha e localização mais específicas de texto, tamanhos de fonte etc.

O que diferencia este trabalho, em relação aos demais, é a aplicação direta de técnicas de PDI e como consequência a geração de uma imagem na saída, contendo apenas a tabela.

III. Estratégia de extração

Com o intuito de oferecer uma solução para a problemática descrita anteriormente, a partir da utilização de várias técnicas, propõe-se a identificação e extração das tabelas. Esta seção descreve as abordagens dessas técnicas.

A Figura 1 demonstra as etapas do processo de extração, o qual é composto por 4 fases:

- Carregamento da Imagem;
- Detecção de Linhas;
- Identificação e Extração da Estrutura de Tabela;
- Geração da Nova Imagem.

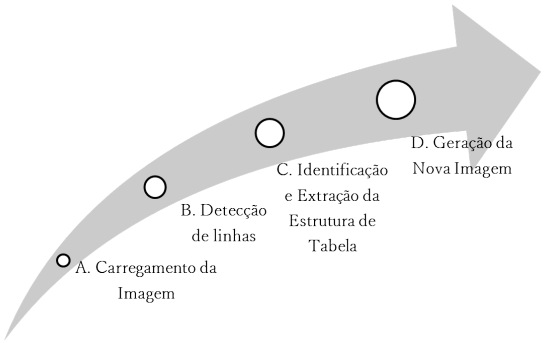


Figura 1. Etapas do processo de extração de tabelas

A. Carregamento da Imagem

A aplicação permite o carregamento (*upload*), por meio do *browser*, de arquivo de imagem com extensão *bmp*, *jpeg* ou *png*, com tamanho máximo de 1MB. O arquivo é enviado por meio do navegador, então é necessário um servidor *web*, optou-se pelo servidor WAMP5 (*Windows Apache MySQL PHP5*). A parte encarregada de efetuar o *upload* do arquivo foi implementado em PHP, já que a verificação dos requisitos de arquivos de imagem admitido é conferido no lado servidor.

B. Detecção de Linhas

De posse da imagem, a verificação da presença de tabela no arquivo de imagem é desenvolvido com a Transformada de Hough – método padrão para detecção de formas geométricas em imagens binarizadas [8].

Antes de aplicar a transformada, a imagem precisa sofrer um pré-processamento, então optou-se por utilizar o algoritmo de Canny para detecção de bordas. O objetivo desse filtro é atenuar ruídos e realçar as bordas da imagem, basicamente destacando o objeto do fundo da imagem [9].

Uma imagem digital é formada por um conjunto de pontos que possuem determinadas características, os quais são chamados de *pixels*. A Transformada de Hough consiste em mapear um *pixel* da imagem em uma curva no espaço de parâmetros – o Espaço de Hough, vide Figura 2 – organizado em forma de um acumulador *n* dimensional, onde *n* corresponde ao número de parâmetros. Quando todos *pixels* tiverem sido processados, é procurado no acumulador os maiores valores (picos). Eles indicam os parâmetros de prováveis linhas (retas) na imagem.

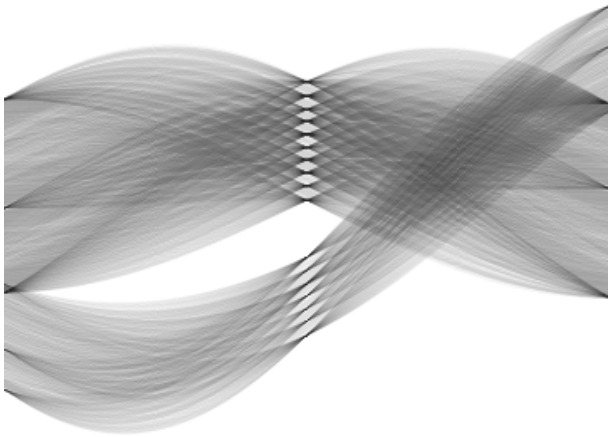
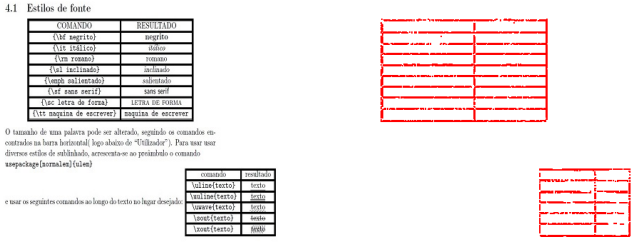


Figura 2. Espaço de Hough

De posse dos picos são desenhada as linhas, formando linhas na horizontal e na vertical, ou seja, gerando uma tabela ou quadro, conforme exemplo da Figura 3. Como as linhas geradas possuem descontinuidades, então aplicou-se técnicas de Morfologia Matemática para fechar as linhas.



(a) (b)

Figura 3. (a) Imagem original (b) Imagem com linhas horizontais/verticais

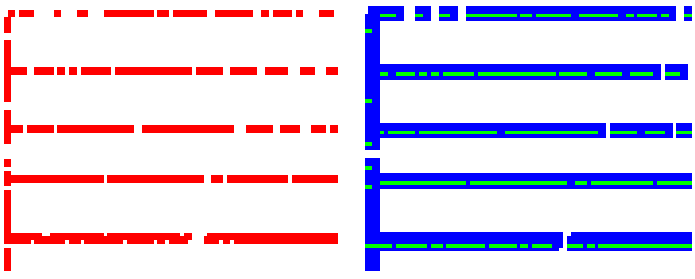
A ideia base da Morfologia Matemática consiste no conjunto de elementos que formam a imagem, os *pixels*, que se agrupam e possuem uma estrutura bidimensional (forma). Certas operações matemáticas, sobre os conjuntos de *pixels*, podem ser usadas para melhorar aspectos específicos das formas como o preenchimento de pequenos buracos e suavização das bordas, para que elas possam ser, por exemplo, localizadas, contadas e reconhecidas nas imagens [10].

Segundo [10], o conceito de morfologia digital é a conectividade de *pixels* adjacentes, cada *pixel* tem um conjunto de vizinhos. O domínio habitual da morfologia são as imagens bi-nível, ou seja, imagens que consistem em apenas de *pixels* pretos ou brancos.

As operações básicas da morfologia digital são a erosão, no qual os *pixels* correspondentes a um determinado padrão são deletados da imagem, e dilatação, em que uma pequena área a cerca de um *pixel* é definido para um determinado padrão [10].

A dilatação, também as vezes chamada de dilação, é uma transformação morfológica que combina dois conjuntos usando adição vetorial. A erosão basicamente encolhe uma imagem e pode ser vista como uma transformação morfológica que combina dois conjuntos usando vetores de subtração [11].

Após aplicação do algoritmo de Dilatação percebeu-se que as linhas ficavam mais fechadas, porém ao se aplicar a dilatação pela segunda vez os resultados ficam melhores, a Figura 4.a exemplifica uma área da tabela com descontinuidades e a Figura 4.b o resultado após a dilatação.



(a) (b)

Figura 4. (a) Imagem gerada após Transformada de Hough (b) Imagem após aplicação de Dilatação

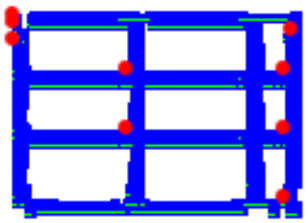
C. Identificação e extração da estrutura de tabela

O algoritmo de Harris [12] é utilizado para identificar os cantos da tabela. Esse algoritmo é baseado na autocorrelação entre os valores digitais da imagem e utiliza derivadas de primeira ordem da função Gaussiana para determinar a magnitude e direção das variações de brilho. Através do cálculo e interpretação geométrica dos autovalores desta função determina a posição do canto na imagem. O grande incremento [13] dado por Harris e Stephens, refere-se ao fato de utilizarem uma máscara de operador Gaussiano tornando o detector menos sensível a ruídos.

A abordagem de Harris para a detecção de bordas faz uso de segmentação de imagens, criando uma nova janela com apenas uma parte da imagem original e deslocando esta imagem em todas as direções sobre a imagem original. Posteriormente, dados dois autovalores para a função de Harris, e calculando a diferença produzida entre a janela e a região que está sendo sobreposta na imagem original, é possível conseguir uma pontuação, onde se define através dela uma constante utilizada para a demarcação de um dos cantos da área definida. Se a pontuação for superior à constante, aquela janela pode ser considerada um canto da área total [14].

O princípio da detecção de cantos reside no fato destes pontos possuírem alto contraste em relação a seus vizinhos. Na prática, os *pixels* de borda detectados, raramente caracterizam completamente uma fronteira ou borda de um objeto, devido ao ruído, a quebra de borda por motivo de iluminação não uniforme e outros efeitos que causam descontinuidades, como visto na Figura 4.a.

Na Figura 5 é possível observar os cantos detectados, em vermelho. Os dados mais extremos, ou seja, o canto superior esquerdo e o canto inferior direito, são identificados como os cantos da tabela e os demais como cantos de célula. Esses dados servirão para a próxima etapa deste projeto.



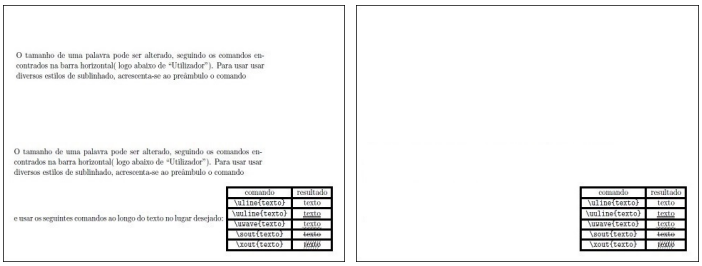
(a)

Figura 5. (a) Imagem original (b) Imagem com a identificação dos cantos

É intuitiva a análise de que, para a identificação dos cantos da tabela poderia-se apenas ter se usado o Algoritmo de Harry e, na fase anterior, não ter sido necessário o Algoritmo de Hough, no entanto o Algoritmo de Harris, nesta sua primeira versão [12], identifica todo e qualquer canto, podendo indicar possíveis cantos, por exemplo, em letras e outras formas geométricas presentes na imagem. Daí a importância da aplicação de Hough, uma vez presente apenas a tabela na imagem a ser processada, o algoritmo de Harris se torna mais eficiente no propósito deste trabalho.

D. Geração da Nova Imagem

A localização identificada, na etapa anterior, é projetada na imagem original, conforme Figura 6.a, então é feito um corte nessa, e os dados recortados são projetados para uma nova imagem, essa contém apenas a tabela identificada durante todo o processo, como na Figura 6.b.



(a) (b)

Figura 6. (a) Imagem original (b) Imagem após o corte

IV. Ferramenta desenvolvida

A ferramenta desenvolvida é composta por páginas feitas em *HTML5*. Na tela principal, o usuário pode dá o *upload* no arquivo de imagem, conforme Figura 7.

Figura 7. Etapas do processo de extração de tabelas

Após o carregamento o usuário seleciona “Aplicar Filtros” ou “Fazer Extração”, conforme Figura 8. Na alternativa filtros há as seguintes opções: cinza, negativo, preto e branco, gaussiano, highpass, laplaciano, roberts, sharpen, saturação, sepia, prewitt e sobel. A Figura 9 exemplifica a aplicação do Filtro de Negativo.

A alternativa extração leva para a página de extração de tabela. Ao selecionar extração, o ambiente aplica a Transformada de Hough e exibe na tela o Espao de Hough e em seguida as linhas em vermelho, destacando a(s) tabela(s) encontrada(s), conforme foi exibido nas Figura 2 e Figura 3. Na maioria das imagens para diminuir a descontinuidade, aplicou-se o algoritmo de dilatação duas vezes. Ao final do processo de extração, a imagem de saída é recortada da imagem original, após definição dos limites das bordas pelo método de Harris, conforme exibido na Figura 6.

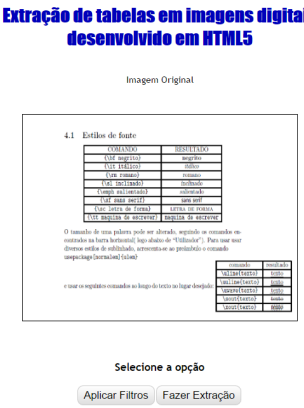
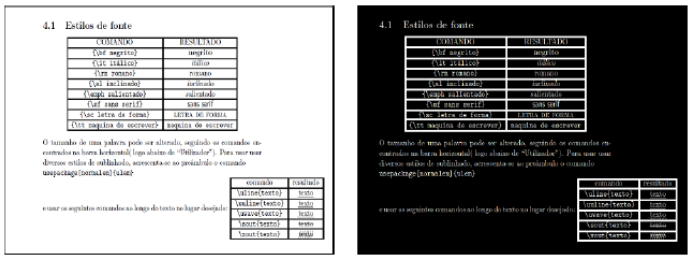


Figura 8. Tela de seleção de opção



(a) Imagem sem filtro (b) Imagem com filtro

Figura 9. Exemplo da aplicação de Filtro Negativo

Durante a validação da ferramenta foi observado que os tipos de imagem (*bmp*, *png*, *jpg*) apresentaram resultados diferentes. A qualidade da imagem gerada, ou seja, as imagens com menos ruídos, em média, foram para as imagens de extensão *bmp*. As imagens de extensão *png* apresentaram resultado razoável. As imagens de extensão *jpeg* foram as que apresentaram uma quantidade maior de ruídos, como exibido na Figura 10.

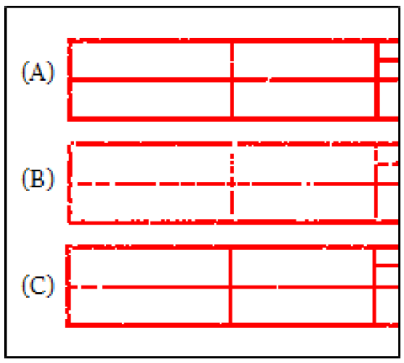


Figura 10. Resultado para formatos (A):*bmp* (B):*jpeg* (C):*png*

Para validar a ferramenta, 10 imagens foram utilizadas, sendo que em apenas 8 haviam uma tabela, a nona possuía duas tabelas e a décima não continha tabela. Ao recortar na imagem original os pontos de extremos, a ferramenta não soube diferenciar no caso de haver duas tabelas, ou seja, falhando em 10% dos casos testados. Para melhorar os resultados, foi adotado o formato de arquivo *bmp* e aplicado o algoritmo de dilatação nas 10 imagens.

V. Conclusão

A ferramenta desenvolvida mostrou-se eficaz, pois ao final do processo, após as etapas de detecção de linhas pela Transformada de Hough, Dilatação, identificação de cantos pelo algoritmo de Harris, foi possível extrair, na imagem original, em 90% dos casos apenas a tabela.

É importante ressaltar que a Transformada de Hough não detecta a forma completa de uma tabela e sim, as várias retas que a compõem. Os algoritmos implementados mostraram-se na maioria das vezes eficientes na detecção das retas, como mostrado nos resultados, porém com alguns ruídos que foram melhorados com filtros de PDI.

O tamanho do arquivo de imagem e/ou o tamanho da tabela (número de linhas e colunas) influencia no tempo de processamento do algoritmo, uma vez que a imagem lida é renderizada *pixel* a *pixel*.

Como retas não são as únicas características que constitui uma tabela, para trabalhos futuros, os autores, planejam a identificação do conteúdo, via OCR, e geração de arquivo no formato de texto, com a tabela e seu conteúdo. Desta forma será possível que o usuário possa manipular a tabela a seu critério.

Referências

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007.

[2] T. Hassan, "Towards a common evaluation strategy for table structure recognition algorithms," in *Proceedings of the 10th ACM symposium on Document engineering*, ser. DocEng'10. New York, NY, USA: ACM, 2010, pp. 255–258. [Online]. Available: http://dl.acm.org/citation.cfm?doid=1860559.1860617

[3] M. Bilauca and P. Healy, "Table layout performance of document authoring tools," in *Proceedings of the 10th ACM symposium on Document engineering*, ser. DocEng'10. New York, NY, USA: ACM, 2010, pp. 199–202. [Online]. Available: http://dl.acm.org/citation.cfm?doid=1860559.1860602

[4] ——, "A new model for automated table layout," in *Proceedings of the 10th ACM symposium on Document engineering*, ser. DocEng'10. New York, NY, USA: ACM, 2010, pp. 169–176. [Online]. Available: http://dl.acm.org/citation.cfm?doid=1860559.1860594

[5] R. K. Padmanabhan, R. C. J, M. Krishnamoorthy, G. Nagy, S. Seth, and W. Silversmith, "Interactive conversion of web tables," in *Proceedings of the 8th international conference on Graphics recognition: achievements, challenges, and evolution*, ser. GREC'09. Springer–Verlag Berlin, Heidelberg, 2009, pp. 25–36. [Online]. Available: http://dl.acm.org/citation.cfm?id=1875535#

[6] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A method to extract table information from pdf files," in *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, ser. IICAI05, 2005, pp. 1773–1785. [Online]. Available: http://ieg.ifs.tuwien.ac.at/projects/pdf2table/

[7] S. Tupaj, Z. Shi, C. H. Chang, and D. C. H. Chang, "Extracting tabular information from text files," in *EECS Department, Tufts University*, 1996. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.3910

[8] P. V. C. Hough, "Method and means for recognizing complex patterns," *U.S. Patent*, 1962.

[9] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4767851&queryText%3DA+Computational+Approach+to+Edge+Detection

[10] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, 2nd ed. Wiley Publishingl, 2011.

[11] J. Facon, *Morfologia Matemtica: Teoria e Exemplos*, Curitiba, Brasil, 1996.

[12] C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.231.1604

[13] M. A. Basso and J. A. S. Centeno, “Estimativa da velocidade de veiculos aplicando o detector de harris,” in *SIMPSIO BRASILEIRO DE SENSORIAMENTO REMOTO*, ser. 16, SBSR, 2013, pp. 2100–2104. [Online]. Available: http://www.dsr.inpe.br/sbsr2013/files/p1552.pdf

[14] L. J. S. Silva, T. B. Mizdal, G. Schardong, L. Netto, A. Frasson, C. Pozzer, E. Passos, and L. Campagnolo, “Arquitetura para o desenvolvimento de um ambiente de simulacao de tiro,” in *Proceedings do XII Simposio Brasileiro de Jogos e Entretenimento Digital*, ser. SBGames 2013, 2013. [Online]. Available: http://www.sbgames.org/sbgames2013/proceedings/comp/06-short-paper-comptrack.pdf