

EasYmage: Uma ferramenta visual para o ensino de Processamento de Imagens Digitais

Inácio C. Alves, Corneli G. Furtado Júnior

Instituto Federal do Ceará (IFCE)
61.939-140 – Maracanaú – CE – Brasil

{inacioalves,cjunior}@ifce.edu.br

Abstract. *The visual stimulus contributes to the teaching-learning process, especially in subjects that requires a high abstraction degree, as the study of Algorithms and Digital Image Processing. In this work, we present EasYmage, a tool build on Colored Petri Nets, wich aims to graphically present the transformation of digital image during execution of classic imaging algorithms. EasYmage eases the specification and validation of imaging algorithms, as well as accelerates the learning curves. As a demonstration, we present an example of an image rotation algorithm.*

Resumo. *O estímulo visual pode contribuir para o processo de ensino aprendizagem, especialmente em tópicos que requerem um alto grau de abstração como o estudo de Algoritmos e Processamento Digital de Imagens (PDI). Neste trabalho, apresentamos o EasYmage, uma ferramenta construída em Redes de Petri Coloridas (RPC), que objetiva apresentar graficamente a transformação de uma imagem digital durante a execução de algoritmos clássicos de PDI. O EasYmage facilita a especificação e validação de algoritmos, bem como acelera a curva de aprendizado. Como uma demonstração, apresentamos o exemplo de um algoritmo de rotação de uma imagem.*

1. Introdução

Imagens digitais são produzidas por uma variedade de dispositivos, tais como câmeras e equipamentos de ultrassom. Elas podem ser utilizadas em aplicações médicas [Dutta and Dubey 2019], militares [Samet et al. 2016], dentre outras. Um dos objetivos da análise de imagens, seja por um observador humano ou por uma máquina, é a extração de informações úteis e relevantes dentro do contexto da aplicação [Gonzalez and Woods 2009].

Diante de sua importância para a sociedade atual, o tópico de “Processamento de Imagens” integra o currículo referência dos cursos de Bacharelado em Ciência da Computação e Informática, da Sociedade Brasileira de Computação (SBC), sendo recorrentemente cobrado no Exame Nacional de Desempenho dos Estudantes (Enade). Segundo [LIMA et al. 2018], nos quatro exames anteriores, quatro por cento das questões foram relacionadas ao tema. Embora não seja um número de questões expressivamente alto quando comparado às áreas predominantes – Algoritmos, Estrutura de Dados e Programação (22%) e Linguagens Formais e Autômatos, Compiladores e Computabilidade (14%) - elas foram classificadas como difíceis e muito difíceis. Zhao

[Zhao et al. 2010] sugere que o baixo índice de acertos se deve: (i) à dificuldade inerente ao tópico Processamento de Imagens, (ii) à dificuldade do aluno na retenção do conhecimento ou (iii) à dificuldade do professor no ensino do assunto. Zhao ainda enumerou algumas dificuldades no ensino de PDI. A primeira refere-se à substituição, devido aos custos, dos experimentos práticos por simples apresentações. A segunda trata do uso de *softwares* comerciais (*Photoshop*, por exemplo) e com grande encapsulamento de funcionalidades, em eventuais experimentos práticos. O encapsulamento esconde do usuário o porquê e como as transformações nas imagens acontecem. Por fim, a terceira dificuldade refere-se ao uso de *softwares* para cálculo numérico (*Matlab*, por exemplo) para a condução das práticas. Segundo ele, embora esses *softwares* disponibilizem funcionalidades convenientes às práticas de Processamento Digital de Imagens (PDI), eles dispõem de uma interface pobre e pouco organizada.

Visando contornar as dificuldades apresentadas por Zhao, o presente trabalho apresenta o EasYmage. Trata-se de um modelo paramétrico, modular e formal descrito em Redes de Petri Coloridas (RPC). Seu objetivo é fornecer um ambiente extensível para as atividades práticas de PDI. Nele, cada operação pode ser apresentada em maior ou menor granularidade, de acordo com a necessidade do aluno ou professor. Para exemplificar o uso do EasYmage e como este pode ser útil ao ensino e aprendizado de PDI, é apresentado passo a passo o processo de espelhamento horizontal de uma imagem didática.

O restante deste trabalho está organizado da seguinte maneira: a Seção 2 apresenta os pré-requisitos necessários ao entendimento dos assuntos envolvidos no presente artigo. Os trabalhos relacionados são apresentados na Seção 3. A Seção 4 apresenta o EasYmage e o experimento de espelhamento horizontal de uma imagem didática. Por fim, a Seção 5 apresenta as considerações finais e perspectivas futuras.

2. Redes de Petri Coloridas

Um sistema a eventos discretos é um sistema dinâmico cuja evolução no tempo é governada pela ocorrência abrupta de eventos físicos, em intervalos possivelmente irregulares. É caracterizado pela mudança discreta de valores de estado; os estados são bem definidos e podem mudar sempre que um evento ocorre [Cassandras and Lafortune 2009].

As RPCs são uma linguagem gráfica e de modelagem para a construção de modelos de sistemas a eventos discretos e análise de suas propriedades. As RPC combinam os recursos das redes Petri (RP) com os recursos de uma linguagem de programação de alto nível. As RPC são destinadas ao uso prático, principalmente porque permitem a construção de modelos compactos e paramétricos. A vantagem de RPC sobre as outras RP é a capacidade de modelar sistemas complexos e fornecer modelos com um alto nível de abstração e melhor representação gráfica [Jensen and Kristensen 2009].

Os modelos de RPC podem conter diferentes níveis de abstração, desde que sejam construídos de forma hierárquica. Cada nível da hierarquia pode conter vários módulos e submódulos que interagem entre si por meio de interfaces bem definidas, de maneira semelhante a linguagens de programação.

As RPC são ferramentas adequadas para a modelagem de operações de PDI, pois fornecem recursos importantes como modularidade, manutenção e expansibilidade. Essas características permitem a adição de novos processos ao modelo original, possibilitando a adição de novas funções ou novos processos do sistema.

Uma RPC é formada por: lugares, transições, arcos, inscrições de arco, fichas coloridas e marcações que juntos permitem a manipulação de dados complexos. A Figura 1 apresenta uma representação de uma RPC.

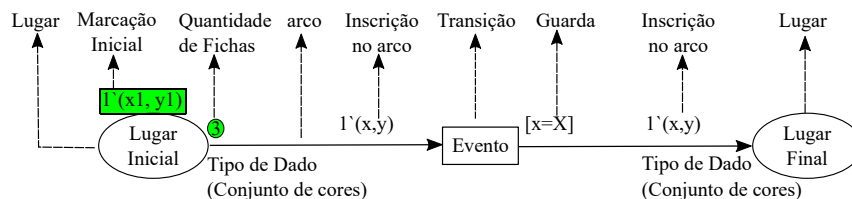


Figura 1. Representação de uma RPC

Lugares são representados por elipses, ou círculos, e possuem um tipo de dado (**conjunto de cores**) associado. **Transições** são denotadas por retângulos. **Fichas**, por pequenos círculos próximos aos lugares e **arcos** por setas. Lugares e transições podem ser nomeados, sendo seus nomes escritos dentro das elipses ou retângulos, respectivamente. Os nomes não possuem significado formal para o modelo, mas ajudam a entender seu funcionamento. Lugares podem ser marcados com fichas, as quais constituem a **marcação do lugar**. Cada ficha representa uma cor (valor) do conjunto de cores (tipo de dados) associado ao lugar. O conjunto de lugares com suas respectivas marcações formam a **marcação atual do modelo** ou **estado atual do modelo**. As marcações e os valores das fichas podem mudar a cada evento.

As transições representam eventos que podem ocorrer no sistema. Para que tais eventos ocorram, é necessário que várias condições sejam satisfeitas. Estas condições são determinadas pelos lugares de entrada (lugares com arco que leva para a transição) das transições, bem como a marcação nesses lugares, as inscrições nos arcos e guardas nas transições (expressão booleana entre colchetes junto à transição).

Quando uma transição ocorre, fichas são removidas dos lugares de entrada e depositadas nos lugares de saída (lugares com arco proveniente de uma transição), respeitando as inscrições nos arcos. Uma **inscrição** é uma expressão escrita junto ao arco, usando a linguagem de programação CPN ML e são constituídas por variáveis, constantes, operadores e funções. Quando todas as variáveis em uma expressão estão ligadas a valores do tipo correto, a expressão pode ser avaliada e a transição pode ocorrer.

Por fim, uma **guarda** em uma transição é uma expressão booleana que deve ser avaliada como verdadeira para que a transição seja habilitada e possa ocorrer. Isto é, uma guarda é uma restrição extra para habilitar uma transição. A descrição formal das RPC pode ser observada em [Jensen and Kristensen 2009].

3. Trabalhos Relacionados

O sentido da visão é responsável pela maior parte das informações que absorvemos. Seja observando um objeto ou lendo um texto, via de regra, são os olhos que nos conectam ao mundo exterior e nos permitem assimilar novos conhecimentos. Por isso os estímulos visuais são tão importantes na sala de aula. Eles não apenas despertam o interesse e a curiosidade dos alunos, mas podem ajudar os estudantes a reterem melhor o conteúdo. Diante disso, diversas são as pesquisas por métodos e ferramentas que facilitem o processo

de ensino-aprendizagem por intermédio do estímulo visual [de Sales Alves et al. 2012], [Lucena et al. 2017], [da Silva Soares et al. 2014] e [Malan and Leitner 2007].

O trabalho de [Caridade 2019] evidencia a importância de experimentos práticos no ensino de conteúdos abstratos e sugere o uso PDI para o ensino de Álgebra Linear. Testes aplicados aos alunos indicam que a metodologia utilizada está próxima ao conceito máximo definido pelo autor (completamente satisfatório).

Nair, Desai e Hebbagil [Nair et al. 2017] propuseram um *web site* para o ensino de PDI, entretanto, esse funciona apenas como uma interface para o *Matlab*. *Matlab*, *Octave* e *Scilab* são *softwares* voltados para o cálculo numérico e integram facilidades para o cálculo com matrizes, processamento de sinais e construção de gráficos. Esses *softwares* podem ser acrescidos de funcionalidades¹ para uma atividade específica. Neles as soluções são descritas como expressões matemáticas, o que pode dificultar para alunos com menor habilidade em matemática. Embora esses *softwares* sejam bastante robustos e especialistas no trabalho de PDI, escondem toda a lógica do processamento de imagens.

Al-Ghaib e Adhami [Al-Ghaib and Adhami 2012] disponibilizaram um curso à distância para o ensino de PDI para alunos de graduação. O curso é composto por dezesseis módulos, contendo exercícios e aplicações práticas. Embora seja um curso introdutório e excelente para apresentar os conceitos básicos sobre o assunto, o uso do *Matlab* como ferramenta de trabalho oculta o funcionamento dos algoritmos de PDI.

Wagner [Wagner et al. 2007] propôs um método para modelar algoritmos de processamento digital de imagens para sistemas distribuídos usando Redes de Petri de alto nível. Embora a proposta seja interessante e tenha relevância, não foi apresentada qualquer implementação de algoritmo para PDI, tampouco foi indicado como os dados que representam imagem e seu processamento são tratados desde a entrada até a saída do modelo.

Ali [Ali et al. 2019] faz uso de uma planilha do *MS-Excel* para o ensino das operações básicas de PDI. Pela planilha, os alunos visualizam e manipulam a matriz de *pixels* de uma imagem, cujas dimensões não podem exceder 256 *pixels* de largura e 256 de altura. No EasYmage, proposto nesse trabalho, foram realizadas simulações em imagens de 640 *pixels* de largura e 480 de altura.

Assim como os trabalhos de [da Silva Soares et al. 2014], [de Sales Alves et al. 2012], [Lucena et al. 2017] e [Caridade 2019], o EasYmage usa o estímulo visual para facilitar o aprendizado de PDI. O resultado das operações sobre matrizes de *pixels* das imagens podem ser visualizados a cada passo de simulação. À medida que o modelo é executado, é possível perceber o movimento dos *pixels* para geração da imagem resultante. Diferente dos softwares *Matlab*, *Octave* e *Scilab* e dos trabalhos propostos por [Du et al. 2015] e [Jianwei et al. 2016], em que os algoritmos de PDI são tratados como uma caixa preta², todos os algoritmos disponíveis no EasYmage são detalhados graficamente. Deste modo, o usuário pode perceber (e visualizar) tudo o que está acontecendo a qualquer instante da execução do algoritmo utilizado.

¹No *Matlab*, a adição de funcionalidades é realizada por extensões conhecidas como *toolboxes*

²Em computação, qualquer ferramenta ou processo que o usuário envia dados e recebe um resultado sem saber como esse resultado foi obtido é denominado de caixa preta.

É importante salientar que o EasYmage foi concebido com fins didáticos. Logo, seu objetivo não é ter performance superior a softwares comerciais como o *Matlab*, e sim facilitar o processo de ensino e aprendizagem de PDI. No EasYmage estão implementados os algoritmos espelhar vertical, espelhar horizontal, girar 90° à esquerda e filtro da mediana (usado na remoção de ruído sal e pimenta), entretanto, é possível a adição e validação de novos algoritmos, conforme apresentado na Seção 4.

Na seção a seguir, são apresentados o EasYmage, um modelo didático e formal para o ensino e validação de algoritmos de PDI baseado em Redes de Petri Coloridas (RPC).

4. O modelo EasYmage

O EasYmage é uma arquitetura para modelagem e simulação de técnicas utilizadas no ensino de PDI, sendo a modelagem baseada no uso de RPC. Operações e cenários podem ser reconfigurados alterando parâmetros do simulador, sem necessidade de modificar a estrutura do modelo.

O modelo RPC do EasYmage é organizado em uma hierarquia de sub-redes (ou páginas). Uma subpágina é um modelo RPC, cuja semântica depende de outras páginas e/ou superpáginas [Jensen and Kristensen 2009]. Essa organização permite a adição de novas operações e, se necessário, a alteração das operações nativamente incorporadas ao modelo.

A Figura 2 apresenta a hierarquia do modelo formal. Como pode ser observado, ele possui um total de sete páginas, divididas em três níveis hierárquicos. As páginas destacadas, em cinza, são apresentadas em detalhes mais adiante.

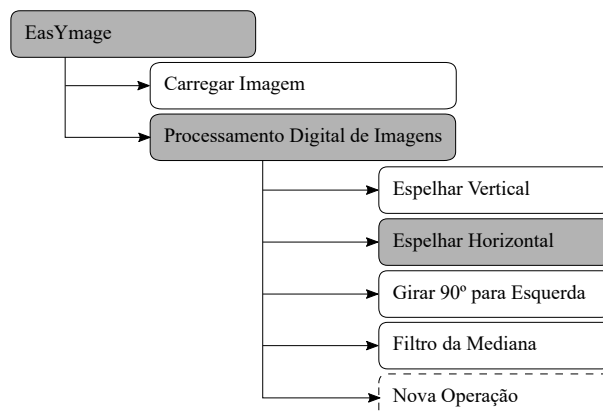


Figura 2. Hierarquia de páginas do EasYmage

A página **EasYmage** apresenta o modelo de processamento de imagens em maior nível de abstração. A página **Carregar Imagem** é responsável pela leitura da imagem do sistema de arquivos e por carregá-la para o EasYmage. Depois de carregada a imagem, na página **Processamento Digital de Imagem** é selecionada a transformação a ser aplicada na imagem original, que resulta na imagem processada. No EasYmage, todas as operações disponíveis são construídas em sub-redes independentes (como subpáginas do modelo “Processamento de Imagens”) para permitir a expansão do modelo de forma simples e sustentável.

As páginas **Espelhar Vertical** e **Espelhar Horizontal** são responsáveis por inverter o sentido da imagem nas direções cima-baixo e direita-esquerda, respectivamente. A página **Girar 90° à Esquerda** é responsável por rotacionar a imagem no sentido anti-horário e, por fim, a página **Filtro da Mediana** substitui o valor de cada *pixel* (x,y) da imagem original pela mediana dos *pixels* da sua vizinhança. Essa operação é útil na remoção de ruídos do tipo sal e pimenta.

A indicação da subpágina **Nova Operação** na Figura 2 reforça a possibilidade de extensão do EasYmage. Novas operações são possíveis pela adição de uma nova transição de substituição no terceiro nível hierárquico do modelo.

A Figura 3 apresenta um exemplo das operações realizadas sobre uma imagem pelas páginas Filtro da Mediana, Espelhar Vertical e Espelhar Horizontal.

A Figura 3(a) apresenta uma imagem original com ruído sal e pimenta. A Figura 3(b) apresenta uma imagem filtrada pela aplicação do filtro da mediana. A Figura 3(c), a Figura 3(d) e Figura 3(e) apresentam, respectivamente, um exemplo das operações “Espelhar Vertical”, “Espelhar Horizontal” e, “Girar 90° à Esquerda” realizadas sobre a imagem filtrada da Figura 3(b).

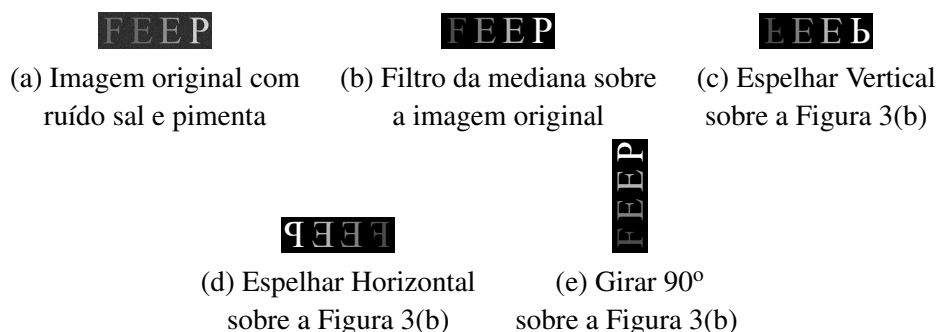


Figura 3. Operações sobre uma imagem

A seguir, nas Subseções 4.1, na 4.2 e 4.3 são apresentadas as sub-redes “EasYmage”, “Processamento Digital de Imagem” e “Espelhar Horizontal”, respectivamente. Essas sub-redes são essenciais para demonstração do caso de uso proposto neste trabalho.

4.1. Rede EasYmage

A Figura 4 apresenta a rede EasYmage. O lugar “Arquivos de Imagens” é associado ao conjunto de cores STRING e modela o caminho para o arquivo de imagem no disco rígido.

Os lugares “Imagem Original” e “Imagem Processada” têm associados o conjunto de cores IMG e modelam a imagem original (pronta para o uso pelo EasYmage, carregada pelo disparo³ da transição “Carregar Imagem”) e a imagem final (criada pela aplicação da operação indicada no lugar “Código da Operação”, após disparo da transição “Processamento de Imagem”).

Por exemplo, o cenário composto por uma ficha de valor “feep.pgm” no lugar “Arquivos de Imagem”, que carrega a imagem da Figura 3(b) para o EasYmage após o

³O mesmo que ocorrência ou execução da ação correspondente à transição

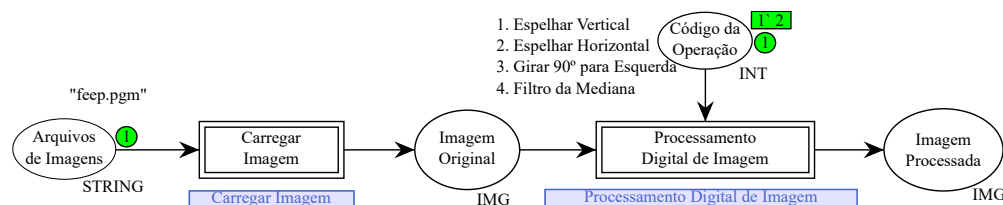


Figura 4. Módulo EasYimage

disparo da transição “Carregar Imagem” e uma ficha de valor “2” no lugar “Codigo da operação”. Após o disparo da transição “Processamento de Imagem”, com a respectiva operação “Espelhar Horizontal” indicada pela ficha de valor “2”, a imagem processada é a mesma da Figura 3(d).

4.2. Sub-rede Processamento de Imagem

A sub-rede “Processamento de Imagem” é apresentada na Figura 5. Nessa sub-rede os lugares “Imagem Original”, “Código da Operação” e “Imagem Processada” são interfaces para a rede EasYimage e têm exatamente o mesmo significado apresentado para aquela rede.

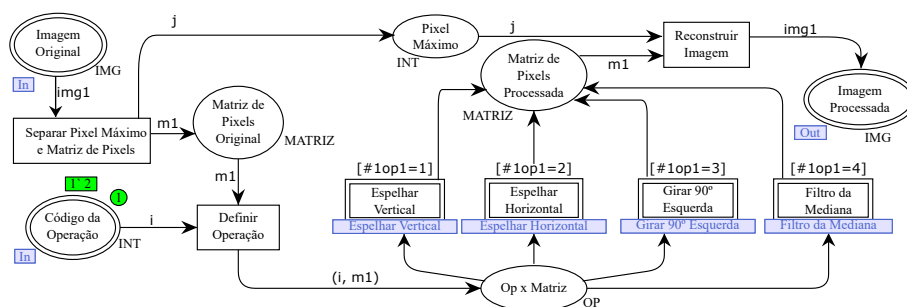


Figura 5. Módulo Processamento Digital de Imagem

O conjunto de cores IMG (dos lugares “Imagem” e “Imagem Processada” são compostas por cinco atributos: identificador do tipo da imagem, largura, altura, intensidade máxima dos *pixels* que a compõem e a matriz de *pixels*. Por exemplo, uma ficha com composição (“P2”, 4, 2, 255, [[0, 204, 153, 255], [255, 102, 51, 0]]) representa uma imagem com as seguintes características: imagem em escala de cinza em ASCII (P2)⁴, largura e altura de 4 e 2 *pixels*, respectivamente, com intensidade máxima de *pixels* 255 e os valores entre colchetes representam as linhas da imagem.

O disparo da transição “Separar *Pixel* Máximo e Matriz de *Pixels*”, deposita no lugar “*Pixel* Máximo” o valor correspondente à intensidade máxima dos *pixels* que compõem a imagem original (que será usado na reconstrução da imagem processada) e a matriz de *pixels* no lugar “Matriz de *Pixels* Original” (a modificação dessa matriz corresponde às operações de PDI).

O lugar “Op x Matriz” recebe fichas do tipo OP, que representam a operação que deve ser realizada sobre a matriz e a própria matriz de *pixels*, e serve de interface com

⁴Dentre os diversos formatos de arquivos de imagens existentes, o presente trabalho usa o formato *Portable GrayMap* por sua simplicidade.

as subredes que realizam o processamento da matriz. Ao final da operação (representada pelo disparo de uma das transições de substituição “Espelhar Vertical”, “Espelhar Horizontal”, “Girar 90° Esquerda” ou “Filtro da Mediana”), a matriz modificada é depositada no lugar “Matriz de *Pixels* Processada”.

Por fim, o disparo da transição “Reconstruir Imagem” reconstrói a ficha do tipo IMG que representa a imagem.

A subseção seguinte apresenta a subrede Espelhar Horizontal e o caso de uso (espelhamento horizontal de uma imagem) proposto neste trabalho.

4.3. Sub-rede Espelhar Horizontal

A Figura 6 apresenta a sub-rede “Espelhar Horizontal”. Nela, as interfaces com a subrede “Processamento de Imagens” é realizada pelos lugares “Op x Matriz” e “Matriz de *Pixels* Processada” e têm exatamente o mesmo significado apresentado para a rede de nível hierárquico imediatamente superior.

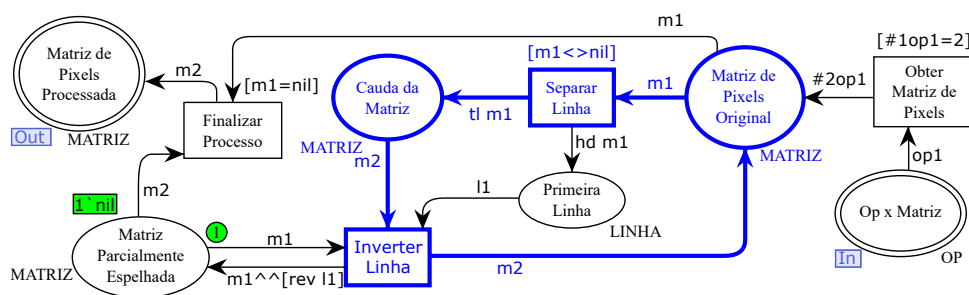


Figura 6. Rede: Espelhar Horizontal

O disparo da transição “Obter Matriz de *Pixels*” deposita a matriz de *pixels* da imagem original no lugar “Matriz de *Pixels* Original”. O disparo da transição “Separar Linha” deposita a primeira linha da matriz da imagem no lugar “Primeira Linha”, e as linhas restantes no lugar “Cauda da Matriz”. Assim, se o lugar “Matriz de *Pixels* Original” tem marcação $[[1, 2], [3, 4], [5, 6]]$, o disparo da transição “Separar Linha” deposita a ficha $[1, 2]$ no lugar “Primeira Linha” e a ficha $[[3, 4], [5, 6]]$ no lugar “Cauda da Matriz”. O disparo da transição “Inverter Linha” inverte horizontalmente os valores que compõem a primeira linha da matriz e a deposita no lugar “Matriz Parcialmente Espelhada”. Assim, caso a marcação do lugar “Primeira Linha” seja $[1\ 2]$, o disparo da transição “Inverter Linha” deposita no lugar “Matriz Parcialmente Espelhada” o valor $[2\ 1]$, que representa a primeira linha da imagem espelhada horizontalmente. Depois, o restante da matriz é depositada no lugar “Matriz de *Pixels* Original” e o processo recomeça até todas as linhas das imagem terem sido destacadas e invertidas para compor a imagem processada.

A Figura 7(a) apresenta uma imagem didática utilizada para exemplificar o processo de inversão horizontal. Após a primeira sequência de disparos das transições “Separar Linhas” e “Inverter Linhas” é obtida a imagem da Figura 7(b). Após a segunda sequência de disparos das mesmas transições é obtida a imagem da Figura 7(c). A Figura 7(d) e a Figura 7(e) apresentam a imagem após a terceira e sétima sequência de disparos das transições “Separar Linhas” e “Inverter Linhas”. Ao final da sétima sequência de disparos, a imagem está completamente processada e pode ser reconstruída para ser exibida ao usuário final.

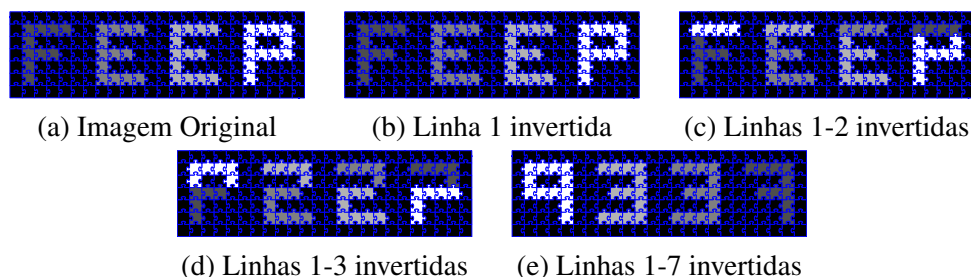


Figura 7. Resultado após as seqüências de disparo das transições “Separar Linhas” e “Inverter Linha”

5. Considerações Finais e Perspectivas Futuras

Nesse trabalho foi apresentado o EasYmage, um modelo formal e paramétrico construído por Redes de Petri Coloridas, que tem por objetivo facilitar o processo de ensino e aprendizagem de Processamento de Imagens Digitais. Por intermédio do EasYmage, os alunos e professores podem visualizar o passo a passo das operações de PDI implementadas no modelo. Estima-se que o estímulo visual possa ajudar o aluno na retenção do conteúdo.

Estão disponíveis no EasYmage as operações Espelhar Vertical, Espelhar Horizontal, Girar 90° a Esquerda, Filtro da Mediana. É importante salientar que novas operações podem ser adicionadas de acordo com a necessidade do usuário. Os cenários de simulação e processamento de imagens são construídos apenas modificando as entradas (fichas) que alimentam o modelo, sendo desnecessário o uso de programação.

Todas as operações construídas no EasYmage seguiram rigorosamente as especificações contidas em [Gonzalez and Woods 2009]. Atualmente estão sendo desenvolvidas as seguintes operações: filtro de Sobel, sépia e média.

Os trabalhos futuros incluem, mas não se limitam a aplicação de curso de PDI usando o EasYmage para os alunos do curso de Bacharelado em Ciência da Computação do Instituto Federal do Ceará (IFCE), aplicação de questionário para verificação do grau de aceitação da ferramenta, e simulado para verificar a aprendizagem. Os resultados dos simulados aplicados aos alunos que fizeram uso do EasYmage devem ser comparados às turmas anteriores, que não usaram a abordagem e os resultados das comparações serão apresentados em novas publicações.

Referências

- Al-Ghaib, H. and Adhami, R. (2012). An e-learning interactive course for teaching digital image processing at the undergraduate level in engineering. In *2012 15th International Conference on Interactive Collaborative Learning (ICL)*, pages 1–5. IEEE.
- Ali, A., Ali, H., and Yousuf, M. A. (2019). Simple image processing in excel. In *2019 IEEE Integrated STEM Education Conference (ISEC)*, pages 46–47. IEEE.
- Caridade, C. (2019). Linear algebra and image processing: a new teaching approach. In *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6. IEEE.
- Cassandras, C. G. and Lafortune, S. (2009). *Introduction to discrete event systems*. Springer Science & Business Media.

- da Silva Soares, M. I., Júnior, C. G. F., Silva, L. C., de Oliveira, R. B., Lima, N. A. S., Soares, É. F., et al. (2014). Visual jo2: Um objeto de aprendizagem para o ensino de programação java a deficientes físicos e auditivos através do estímulo visual—um estudo de caso. *RENOTE-Revista Novas Tecnologias na Educação*, 12(2).
- de Sales Alves, R., de Araujo, J. O. A., and Madeiro, F. (2012). Alfabetokinet: Um aplicativo para auxiliar na alfabetização de crianças com o uso do kinect. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*.
- Du, Y., Gai, L., Tian, J., and Liu, W. (2015). Digital image processing teaching auxiliary system based on matlab graphical user interface. In *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, pages 434–438. IEEE.
- Dutta, A. and Dubey, A. (2019). Detection of liver cancer using image processing techniques. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, pages 0315–0318.
- Gonzalez, R. C. and Woods, R. C. (2009). *Processamento digital de imagens*. Pearson Educación.
- Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri nets: modelling and validation of concurrent systems*. Springer Science & Business Media.
- Jianwei, Z., Jifang, Z., Tao, C., Zhaohui, L., and Qingtuan, L. (2016). Exploration of the innovative medical talent training based on problem-based learning in digital image processing teaching. In *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, pages 163–167. IEEE.
- LIMA, P. D. S., Ambrósio, A. P., Brancher, J. D., and Felix, I. (2018). Sysenade-análise das questões de provas do enade organizadas pelos temas abordados. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, page 419.
- Lucena, G., Cidrim, L., and Madeiro, F. (2017). Um aplicativo para estimulação da memória visual em crianças disléxicas por meio do jogo de palavras cruzadas. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, page 685.
- Malan, D. J. and Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM Sigcse Bulletin*, 39(1):223–227.
- Nair, A., Desai, M., Hebbagil, P., and Lade, N. (2017). Interactive e-learning for digital image processing. *International Journal of Engineering Research Technology (IJERT)*, 5.
- Samet, R., Bayram, A., Tural, S., and Aydin, S. (2016). Primer defects detection on military cartridge cases. In *2016 Nicograph International (NicoInt)*, pages 96–99.
- Wagner, B., Dinges, A., and Muller, P. (2007). Dataflow orchestration of image processing algorithms using high-level petri nets. In *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*, pages 344–347.
- Zhao, H., Tang, J., and Luo, B. (2010). Teaching reform and innovation of the course—digital image processing experiments. In *2010 5th International Conference on Computer Science & Education*, pages 1599–1600. IEEE.