

COMPARAÇÃO DE ALGORITMOS DE PROCESSAMENTO DIGITAL DE IMAGENS APLICADAS À RESOLUÇÃO DE LABIRINTOS BIDIMENSIONAIS

João Henrique Laranja Capucho¹
Karin Satie Komati²

Resumo

Um labirinto é uma rede de caminhos e paredes que é representado como um quebra-cabeça onde se tem que encontrar um caminho do início ao ponto de destino. Este trabalho pretende avaliar as abordagens de solução de labirintos baseado em métodos de Processamento Digital de Imagens, Watershed, Esqueletização e Morfologia Matemática. A comparação visa compreender os pontos positivos e negativos de cada técnica e a qualidade dos resultados obtidos. Foram três diferentes experimentos: (i) em labirintos perfeitos, (ii) em labirintos com duas rotas de saída e (iii) em labirinto complexo. Os resultados do estudo mostram que para o problema de resolução de caminho, a Morfologia Matemática obteve resultados significativamente superiores em relação aos outros métodos estudados.

Palavras-chave. Processamento Digital de Imagens. Watershed. Esqueletização. Morfologia Matemática.

COMPARISON OF DIGITAL IMAGE PROCESSING ALGORITHMS APPLIED TO RESOLUTION OF TWO-DIMENSIONAL MAZES

Abstract

A maze is a network of paths and walls represented as a puzzle where one must find a path connecting the start of the maze to its finish. This work's goal is to evaluate the different approaches for solving mazes based on Digital Image Processing, Watershed, Skeletonization, and Mathematical Morphology methods. The study aims to gather knowledge about the advantages and disadvantages of each method, and about the quality of their results. Three different experiments were conducted: (i) with perfect mazes; (ii) with mazes with more than one exit route; and (iii) with complex mazes. Our work's results show that, for the problem of path finding, the Mathematical Morphology methods obtained significantly superior results when compared to the other methods.

Keywords. Digital Image Processing. Watershed. Skeletonization. Mathematical Morphology.

1 Coordenação de Informática, Instituto Federal do Espírito Santo (Ifes) Campus Serra, jhcapucho@gmail.com

2 Coordenação de Informática, Instituto Federal do Espírito Santo (Ifes) Campus Serra, kkomati@ifes.edu.br

1. INTRODUÇÃO

Robôs autônomos são robôs que podem realizar tarefas de forma inteligente, sem qualquer assistência humana. "Maze Solving Robot", também chamado de "Micro-Mouse Robot", é um dos robôs autônomos mais populares existentes. É um pequeno e autossuficiente robô que pode resolver um labirinto de uma posição de partida conhecida até o centro da área do labirinto ou até a saída, no menor tempo possível (vide Figura 1.1) (AWHAD et al., 2010). Competições Micro-Mouse existem há cerca de três décadas e continuam sendo consideradas uma importante questão da área de robótica (ALSUBAIE, 2013). Uma foto de uma competição de Micro-Mouse nos EUA é apresentada na Figura 1.2, que mostra um labirinto de 16x16 células.

O conceito da competição é colocar o robô em um ambiente desconhecido onde ele deve cumprir uma tarefa preestabelecida, exigindo capacidade de tomada de decisão. Para tal, é importante que o robô seja capaz de fazer a exploração do ambiente (do labirinto) e que à medida de sua navegação seja capaz de construir um mapa do ambiente. Dado que há um mapa, o próximo passo é analisá-lo para atingir o objetivo (MARKOSKA et al., 2015).

Um labirinto é uma área bidimensional do tipo grid de qualquer tamanho, geralmente retangular, e consiste em células (espaço delimitado). Uma célula é um espaço elementar e, de forma simplificada, é possível visualizar uma célula como uma casa de um tabuleiro de xadrez. Um labirinto é uma rede de caminhos e paredes que é representado como um quebra-cabeça onde se tem que encontrar um caminho do início ao ponto de destino. O labirinto contém diferentes tipos de dificuldades, sendo que a sua complexidade é determinada pelo número de células, de paredes, de corredores, pela distância entre o início-término na estrutura de labirinto e pela quantidade de diferentes rotas para a saída.

Este trabalho pretende avaliar as abordagens de solução de labirintos baseado em técnicas de Processamento Digital de Imagens (PDI) (GONZALEZ; WOODS, 2006). PDI é uma área de destaque na computação devido à sua capacidade de viabilizar categorias distintas de aplicações. Várias são as aplicações de PDI, entre as quais: aplicações de filtros em imagens de satélite de modo a destacar as

áreas em processo de desmatamento; já na área médica, o realce das imagens obtidas através de ressonâncias ou exames de imagens tridimensionais, de forma a facilitar a visualização dos pontos mais importantes para o médico.

Figura 1.1. Foto de um Micro-Mouse IEEE. Fonte: Lessard, Ronald A. (1998)

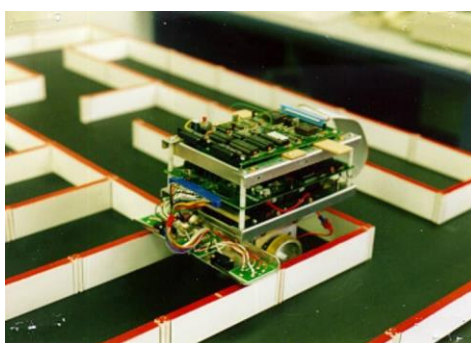
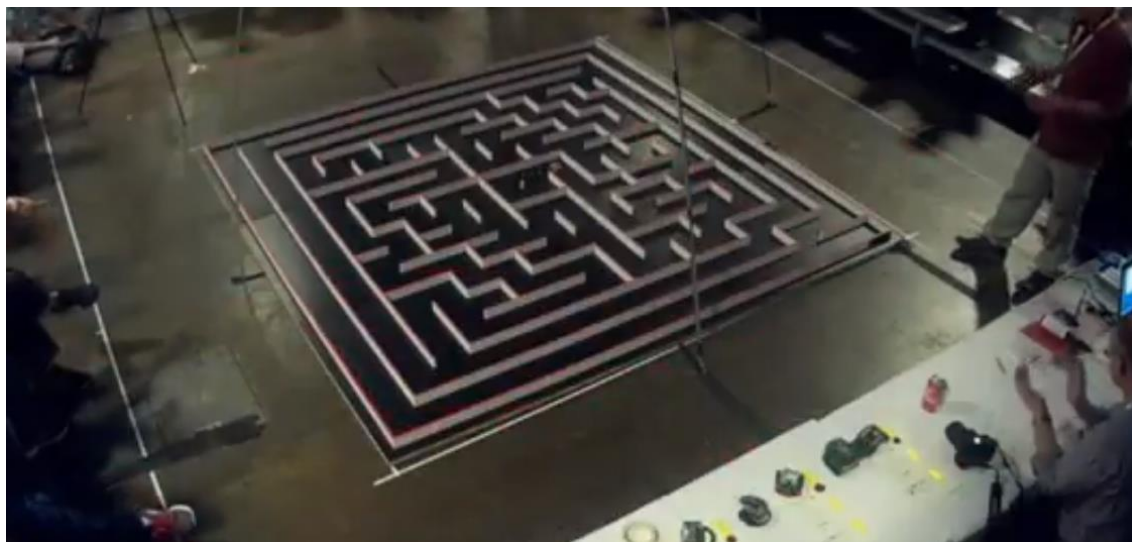


Figura 1.2. Foto da competição Micro-Mouse nos EUA em 2009 da IEEE. Fonte: APEC 2017 (<https://www.youtube.com/c/MicromouseMeUk/>) sob licença Creative Commons do YouTube.



O trabalho tem como finalidade comparar três soluções baseadas em métodos de PDI para o processamento da imagem de um labirinto bidimensional e apresentar o seu respectivo caminho de saída. Os métodos abordados serão: Watershed, Esqueletização e Morfologia Matemática.

Existem vários algoritmos para solução de labirintos, tais como: Seguidores de Parede (*Wall Follower*) e o método de Tremaux. No entanto, segundo Singh e Sekhon (2011) essas são estratégias ruins, pois não há garantia de solução mesmo quando existe uma. Outras formas de solução envolvem estratégia de busca pelo melhor caminho via Dijkstra ou A* (MURATA; MITANI, 2014), ou abordagem via Algoritmos Genéticos (SINGH; PANDEY, 2014) ou até mesmo via Autômatos Celulares (CHAUDHARY; GARG, 2013).

O trabalho de Kathe et al. (2015) também usa a técnica de Esqueletização, mas apresentando os resultados com apenas duas imagens de testes. Uma técnica similar ao Watershed é o Flood-Fill, que é usado no estudo de caso do trabalho de Elshamarka e Saman (2012), apresentando resultados de várias execuções de um Micro-Mouse em um único labirinto. Não se encontrou trabalho correlato que utiliza Morfologia Matemática.

1.1. REFERENCIAL TEÓRICO

Compreende-se a imagem digital como uma matriz onde as respectivas coordenadas de seus eixos identificam um ponto na imagem, que possui como valor o seu nível na escala de cinza (no caso das imagens em tons de cinza). Assim, uma imagem digital pode ser vista como uma representação bidimensional com um conjunto limitado de valores digitais conhecidos como "pixels" (PICTure ELEMENT) (MARQUES FILHO; VIEIRA NETO, 1999). Uma imagem I pode ser definida como:

$$I = f(x, y), \quad (\text{Eq. 1})$$

onde $f(x, y)$ é uma imagem digital e f é uma função que atribui um valor na escala de cinza para cada par distinto de coordenadas (x, y) (GONZALEZ; WOODS, 2006). Por exemplo, na Figura 2.1, cada posição da matriz é o resultado da aplicação de uma função $f(x, y)$, e esse valor representa a intensidade da cor do pixel naquela posição.

Figura 2.1. Representação de uma imagem digital.

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix}.$$

PDI basicamente trata de operações em matrizes, o que permite aplicar operações aritméticas e lógicas nos valores dessa matriz, possuindo a seguinte estrutura:

$$A \text{ operação } B = Z \quad (\text{Eq. 2})$$

O resultado dessa equação seria uma imagem cujos pixels correspondem ao resultado da *operação* aplicada para cada elemento de A e B. Assim, para entender as operações, essas podem ser decompostas em dois grupos: operações aritméticas (soma, subtração, multiplicação e divisão) e operações lógicas (and, or, xor, not).

Há várias operações básicas de PDI, sendo que uma delas é a binarização, que consiste na transformação de uma imagem em escala de cinza em uma imagem binária ou *Black-and-White*, isto é, com apenas dois valores. Definindo-se um limiar, todos os valores acima dele são considerados de cor branca e todos os valores abaixo dele, de cor preta.

Uma propriedade importante em uma imagem é a relação entre os pixels, como por exemplo, a conectividade entre eles. Para se estabelecer se dois pixels estão conectados, é necessário determinar se eles são adjacentes segundo algum critério (um exemplo é pixels 8-conectados, ou conectividade-8, que são todos os vizinhos de um pixel) e se seus níveis de cinza satisfazem a um determinado critério de similaridade. Avaliando uma sequência de pixels que estão conectados é possível se definir um caminho conectado.

1.2. MORFOLOGIA MATEMÁTICA

O termo Morfologia, segundo Parker (2010), significa a forma e a estrutura de um objeto, ou até mesmo os arranjos e as inter-relações entre as partes de um objeto. Para a definição das operações de Morfologia Matemática em imagens binárias, considere que dois conjuntos $\{A, B\} \in Z$, e que possuem componentes $a = (a_1, a_2)$ e $b = (b_1, b_2)$. Assim, as seguintes operações podem ser definidas:

- Translação de A por $x = (x_1, x_2)$: $(A)_x = \{c \mid c = a + x, \text{ para } a \in A\}$
- Reflexão de B: $B' = \{x \mid x = -b, \text{ para } b \in B\}$.
- Dilatação: $A \oplus B = \{x \mid B'_x \cap A \neq \emptyset\}$
- Erosão: $A \ominus B = \{x \mid B_x \in A\}$.

Os operadores de dilatação e erosão funcionam através da passagem de dois parâmetros ao operador, a imagem de entrada a ser dilatada e um conjunto de coordenadas, conhecido como elemento estruturante, que consiste em um padrão especificado como as coordenadas de um número discreto de pontos relativos a alguma origem. É esse elemento que determinará o efeito preciso da operação na imagem resultante. Esses operadores são bastante utilizados em diversas aplicações de processamento digital de imagens, principalmente no que tange à geometria e topologia da imagem, como por exemplo, detecção de bordas.

A dilatação de A por B é o conjunto de todos os x deslocamentos para os quais a intersecção de B' e A inclui pelo menos um elemento diferente de zero. De forma simplificada, o efeito básico do operador em uma imagem binária é expandir gradualmente as bordas de regiões em primeiro plano (regiões em que os pixels são tratados em lógica booleana como verdadeiro). Em contrapartida, os buracos presentes nessas regiões têm sua área reduzida na imagem.

O processo de erosão resulta no conjunto de pontos x tais que B, transladado de x, está contido em A. O efeito básico deste operador em uma imagem binária é realizar a erosão de regiões de bordas dos pixels em primeiro plano. Assim, áreas de pixels em primeiro plano reduzem de tamanho, e os buracos nessas áreas se expandem.

Figura 2.2. Aplicação da operação de dilatação em imagem binária. Fonte: Imagem retirada e adaptada de Fisher et al. (2003).

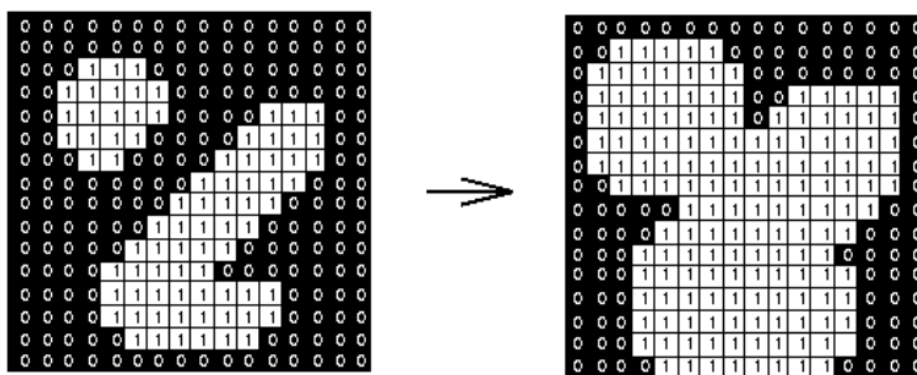
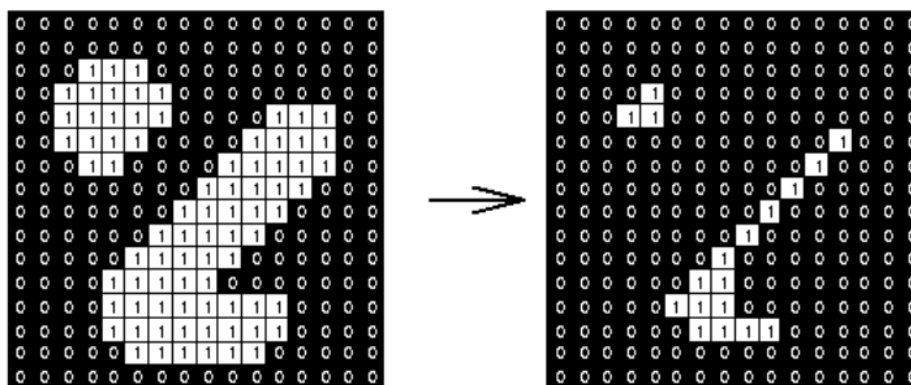


Figura 2.3. Aplicação da operação de erosão em imagem binária. Fonte: Imagem retirada e adaptada de Fisher et al (2003).



1.3. WATERSHED

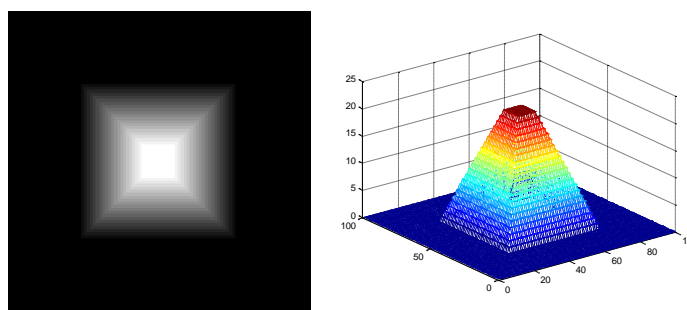
A transformação Watershed foi introduzida como uma técnica para segmentação de imagens em tons de cinza por S. Beucer e C. Lantuéjoul no final dos anos 1970 e atualmente é utilizada como um passo fundamental em procedimentos de segmentação (COUPRIE; BERTRAND, 1997).

Todavia, o termo é proveniente de outra área de conhecimento, a geografia. Define-se como Watershed um cume que divide áreas drenadas por diferentes sistemas fluviais e possui bacias

hidrográficas que são áreas geográficas que drenam para um rio ou outro corpo de água (LIU et al., 2012). Portanto, segundo Liu e colegas (2012), se uma imagem em tons de cinza pudesse ser vista como uma superfície topológica, na qual os tons de cinza de cada pixel representam a altura da superfície, as bacias resultantes definem o que seria a segmentação da imagem. Este é um processo bastante utilizado, por exemplo, em segmentação de rodovias e separação de grãos, como café.

Dessa maneira, considerando um realce topográfico de uma imagem, as áreas mais escuras representariam as bacias e os vales, enquanto as montanhas e as linhas de crista corresponderiam às áreas mais claras. Na Figura 2.4 apresenta-se um exemplo de uma imagem 2D e de sua representação 3D, na qual a área branca na imagem 2D é a região mais alta e em vermelho na imagem 3D, e a região das bordas em preto na imagem 2D é a parte da base na cor azul na imagem 3D.

Figura 2.4. Imagem de entrada e sua visualização topográfica em 3D. Fonte: Elaborada pelo autor.

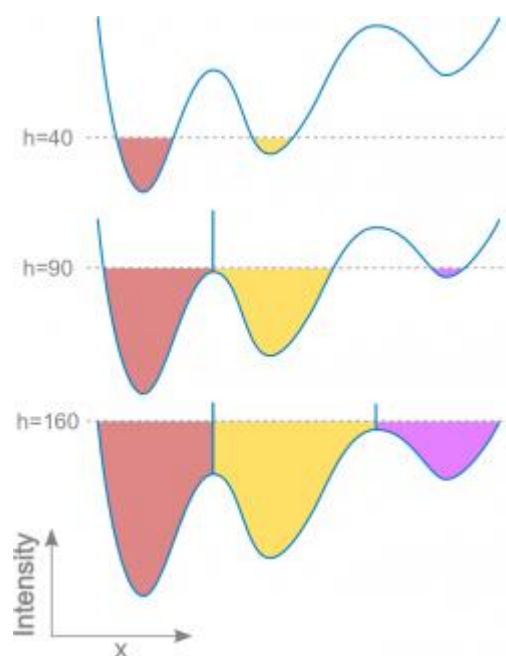


Assim, seguindo essa interpretação topográfica do procedimento, três tipos de pontos podem ser considerados: os que pertencem a um mínimo regional; pontos nos quais uma gota d'água, se posicionada na localização de quaisquer desses pontos, certamente cairia em um único mínimo, e pontos nos quais a água teria chances igualmente prováveis de cair para mais de um desses mínimos. Para um mínimo local particular, o conjunto de pontos que satisfazem a segunda condição são os chamados de bacias. Os pontos que satisfazem a terceira condição formam as linhas de crista e são denominados *Watershed Lines* (GONZALEZ; WOODS, 2006).

Para o entendimento do processo, é possível visualizar a seguinte abordagem: imagine a paisagem sendo imersa em um lago, com perfurações em mínimos locais. A partir desses mínimos locais, bacias

serão preenchidas com água e, em pontos onde a água proveniente de diferentes bacias se encontrariam, as barragens são construídas. Quando o nível da água atingir o pico mais alto da paisagem, o processo é interrompido, pois constroem-se barreiras quando diferentes fontes de água se encontram. Um exemplo esquemático do processo é mostrado na Figura 2.5, na qual foi feito um corte no eixo da altura, mostrando três vales. À medida que a água sobe, enche os vales e, no momento em que duas fontes de água se encontram, as barreiras são construídas. Como resultado, a paisagem é dividida em regiões ou bacias separadas por barragens (barreiras), chamadas de Watersheds. A página de Beucher³ possui imagem com simulação passo a passo do processo.

Figura 2.5. Visão esquemática do processo de Watershed em 1D. Fonte: Imagem retirada de https://imagej.net/Classic_Watershed



3 <http://cmm.ensmp.fr/~beucher/wtshed.html>

Revista Ifes Ciência, v. 3, nº 2, 2017 – Instituto Federal do Espírito Santo.

1.4. ESQUELETIZAÇÃO

Através do processo de afinamento das bordas em uma imagem digital é possível se obter uma forma simplificada da imagem, utilizando um processo que removerá os pixels que não possuem informações relevantes, resultando em uma imagem com todas as suas características essenciais, todavia, simplificada. Esse processo possui o nome de Esqueletização (conhecido também por Esqueletonização - Skeletonization - ou Afinamento - Thinning). Sendo assim, o resultado deste processo é conhecido como “esqueleto” da imagem digital (PLOTZE; BRUNO, 2004). Este processo pode ser utilizado em diversas aplicações, como detecção de digitais, reconhecimento de caracteres e de assinaturas e até mesmo na área médica, simplificando imagens geradas pelos equipamentos de imagem, tornando mais fácil a avaliação do resultado.

Há várias propostas de implementação de Afinamento, tais como: Zhang Suen, Holt, Stentiford e Hilditch (PARKER, 2010). Para este trabalho, foi utilizado o algoritmo de Hilditch. A proposta de Hilditch é um algoritmo iterativo, que tem como base a mudança de cor do pixel de preto para branco fundamentado em um conjunto de regras. Para poder explicar o algoritmo, é necessário definir o número de transições de um pixel e o número de vizinhos associados a ele.

Considere que P_1 é o pixel central de uma matriz 3×3 , e que $A(P_1)$ é o número de transições de zero para um na sequência de P_2, P_3, \dots, P_9 (conforme Tabela 2.1, mostrando o pixel central e seus 8 vizinhos) e $B(P_1)$ representa o número de vizinhos do pixel central diferentes de zero. Portanto, a cada passada do algoritmo os elementos que satisfaçam as regras da Tabela 2.2, na ordem mostrada, devem ser removidos (PLOTZE; BRUNO, 2004). O processo será interrompido quando todo o conjunto de elementos for varrido sem que nenhum pixel seja removido.

Na primeira regra, o fato de ser maior ou igual ao valor 2 significa que nenhum pixel que seja ponto final possa ser removido (qualquer elemento com um vizinho em tom preto é um pixel de ponto final). Já a condição de que seja menor ou igual a 6, garante que o pixel pertença à borda de uma região.

Tabela 2.1. Ordem dos pixels relativos a um pixel central P1.

P9	P2	P3
P8	P1	P4
P7	P6	P5

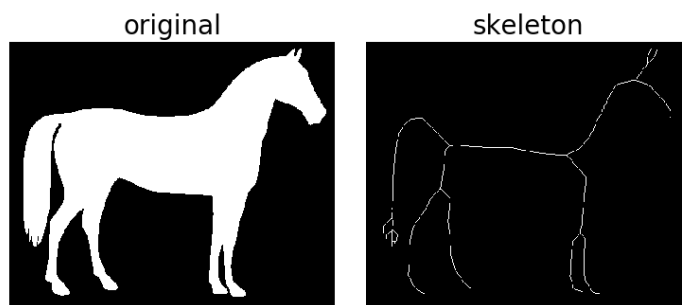
Tabela 2.2. Conjunto de regras utilizadas no algoritmo de Hildith

1	$2 \leq B(P1) \leq 6$
2	$A(P1) = 1$
3	$(P2 \text{ ou } P4 \text{ ou } P8) \text{ ou } A(P2) \neq 1$
4	$(P2 \text{ ou } P4 \text{ ou } P6 = 0) \text{ ou } A(P4) \neq 1$

A segunda regra da listagem tem como objetivo testar a conectividade: caso o pixel P1 seja removido, fará com que seus vizinhos se tornem desconexos. A terceira regra faz com que o algoritmo garanta que linhas verticais de dois pixels de largura não sofram erosão completa pelo algoritmo. Já a quarta regra faz com que o algoritmo garanta que linhas horizontais de dois pixels de largura não sofram erosão completa pelo algoritmo.

Um exemplo do resultado desta técnica é mostrado na Figura 2.6, à esquerda é a imagem de entrada e à direita é sua respectiva imagem resultante do processo de Esqueletização. É possível verificar que a imagem esqueleto preserva a forma do cavalo, mas através de linhas com espessura de um pixel.

Figura 2.6. Imagem de entrada e resultado após a aplicação da Esqueletização. Fonte: Imagem retirada de http://scikit-image.org/docs/0.13.x/auto_examples/edges/plot_skeleton.html



2. MATERIAIS E MÉTODOS

Para o desenvolvimento dos três sistemas foi utilizado o ambiente de desenvolvimento MATLAB® e uma biblioteca de PDI, a DIPlib (VLIET, 2003).

SISTEMA COM WATERSHED

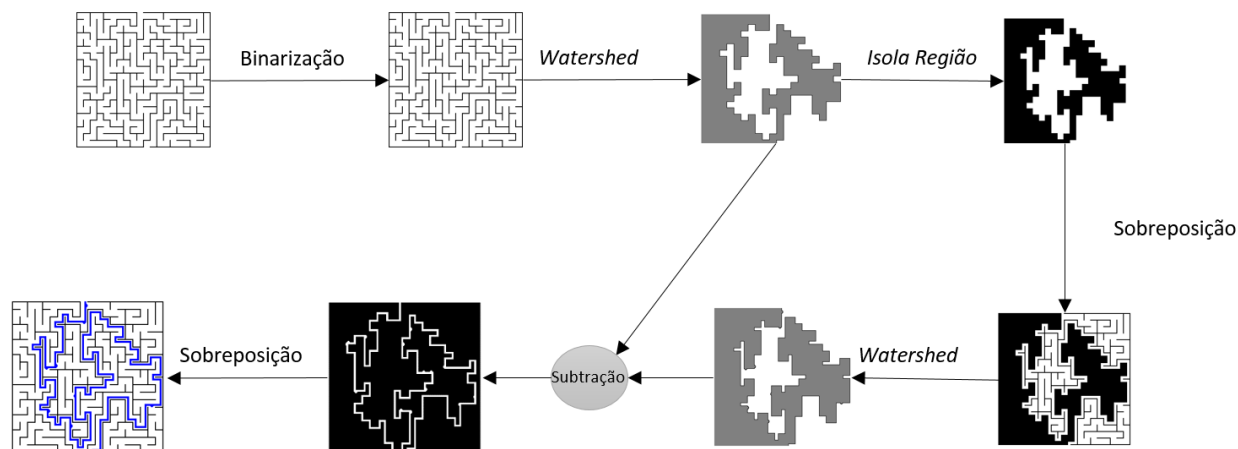
A imagem da Figura 3.1 apresenta o algoritmo geral do processamento, exibindo os resultados das etapas principais executadas (MATHWORKS, 2014). A primeira etapa é a conversão da imagem de entrada para uma imagem preto e branco. Assim, o resultado é uma matriz bidimensional composta por apenas dois valores, zeros e uns.

Logo em seguida, é aplicada a transformada Watershed (MATHWORKS, 2016), que retorna uma matriz que identifica as regiões de Watershed da matriz de entrada. Os elementos da matriz de saída da transformada são valores inteiros maiores ou iguais a zero. Os elementos valorados em zero não pertencem a quaisquer regiões Watershed. Os elementos de valor um pertencem à primeira região, os de valor dois à segunda região e assim por diante. Assim, o processo “Isola Região”, converte os valores para duas regiões, tal como uma imagem binária (Figura 3.2a).

Sobrepõe-se este último resultado na imagem de entrada, na qual a região branca é substituída pela imagem de entrada. É esta parte, onde se é possível visualizar parte do labirinto original, que

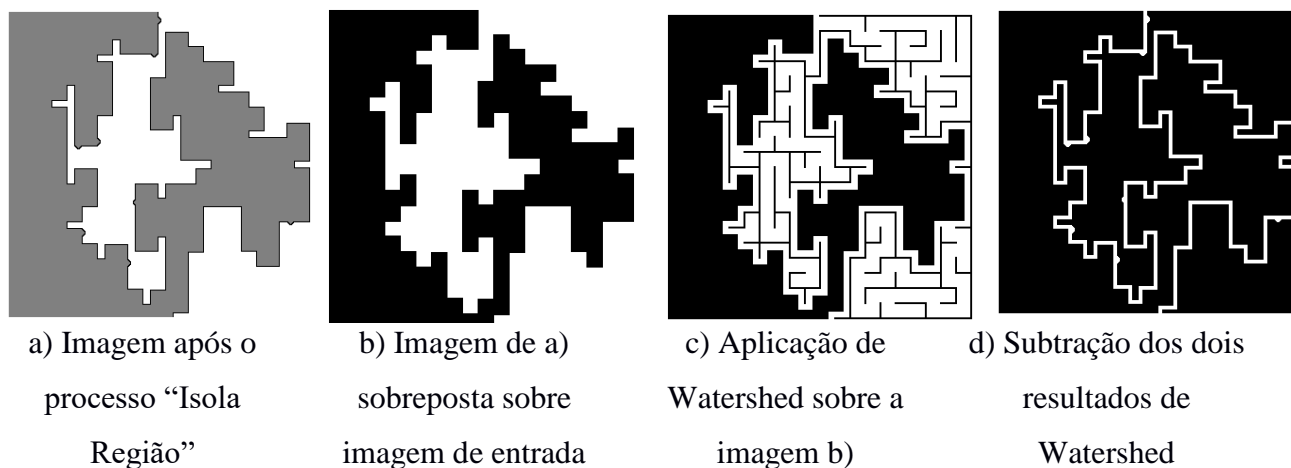
contém o caminho de saída. A Figura 3.2b mostra o resultado dessa aplicação. Aplica-se novamente a transformada Watershed nesta imagem sobreposta, cujo resultado é mostrado na Figura 3.2c.

Figura 3.1. Arquitetura do processamento utilizando Watershed. Fonte: Elaborada pelo autor.



A imagem resultante da segunda aplicação de Watershed possui apenas uma diferença da imagem resultante da primeira aplicação Watershed, que é a porção da imagem que representa exatamente o caminho de saída do labirinto. Desta forma, é possível simplesmente subtrair a primeira imagem da segunda, para que se obtenha uma imagem como caminho de saída, como na Figura 3.2d. Este resultado depois é sobreposto na cor azul sobre a imagem de entrada apenas para facilitar a visualização do usuário.

Figura 3.2. Imagens intermediárias do processo via Watershed. Fonte: Elaborada pelo autor.



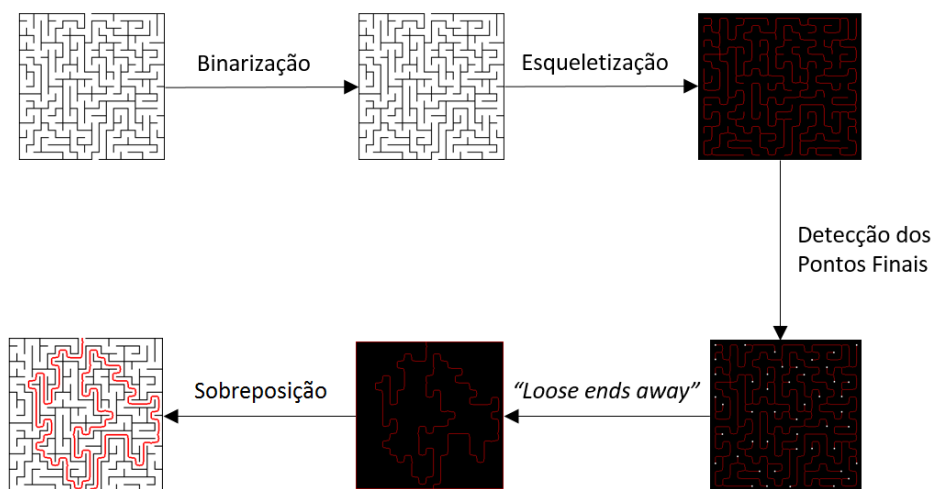
2.1. SISTEMA BASEADO EM ESQUELETIZAÇÃO

A arquitetura utilizada foi replicada da proposta feita por Cris Luengo (LUENGO, 2010). A Figura 3.3 apresenta o algoritmo geral do processamento, exibindo os resultados das etapas principais executadas pelo método estudado.

Primeiramente, um ponto relevante a ser considerado é uma diferença essencial no processamento desta técnica em relação à anterior. A primeira técnica apresentada trabalhava as paredes do labirinto com o objetivo de encontrar o caminho do labirinto. Através das bordas das regiões, era possível identificar a rota de saída. Já a Esqueletização propõe uma solução diferente, em vez de trabalhar com as paredes, o sistema via Esqueletização apresentará um resultado com todos os caminhos possíveis dentro do labirinto.

Inicialmente, a imagem de entrada é binarizada e aplica-se a técnica de Esqueletização (Figura 3.4a). É possível observar que todos os caminhos possíveis do labirinto são apresentados nesta imagem. As linhas em vermelho representam as partes brancas entre as paredes do labirinto da imagem de entrada, porém, afinadas. A cor vermelha é mostrada apenas para facilitar a visualização do leitor, pois como a imagem é binária só possui dois valores, o vermelho equivale ao 1 (um) e o fundo ao 0 (zero).

Figura 3.3. Arquitetura do processamento utilizando Esqueletização. Fonte: Elaborada pelo autor.

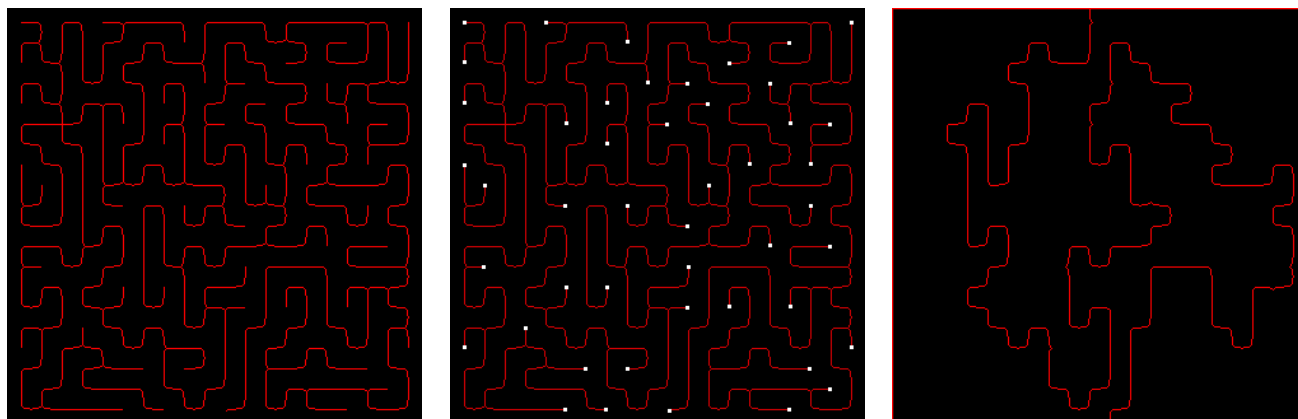


O objetivo dos próximos passos do algoritmo é tratar essa imagem de forma a remover todos os pixels que não pertencem ao conjunto de pixels da rota de solução do labirinto. Para atingir esse objetivo, o primeiro passo é a detecção de todos os pontos que representam o final de um caminho existente no labirinto. Um pixel é considerado um pixel final caso ele tenha apenas um vizinho de valor 1 (um), ou seja, seu único vizinho seja o pixel anterior a ele. Dessa forma, é possível identificar todos os locais que podem ser uma possível entrada ou saída do labirinto. A Figura 3.4b apresenta os pontos finais (em branco) encontrados no resultado da aplicação da Esqueletização.

O DIPLib fornece uma função que faz com que todos os pontos finais sejam removidos, iterativamente, até que não reste mais pontos finais a serem retirados. Restando, assim, os caminhos que possuem uma conexão com a borda da imagem, ou seja, resta o caminho que forma a rota de saída do labirinto (Figura 3.4c).

Sobrepõe-se esse resultado na imagem inicial do labirinto utilizado como entrada, para que o algoritmo exiba a rota de saída de forma mais compreensível. Aplicou-se a operação de dilatação na rota apenas para que essa ficasse mais grossa e portanto mais visível no resultado final.

Figura 3.4. Imagens intermediárias do processo via Esqueletização. Fonte: Elaborada pelo autor.

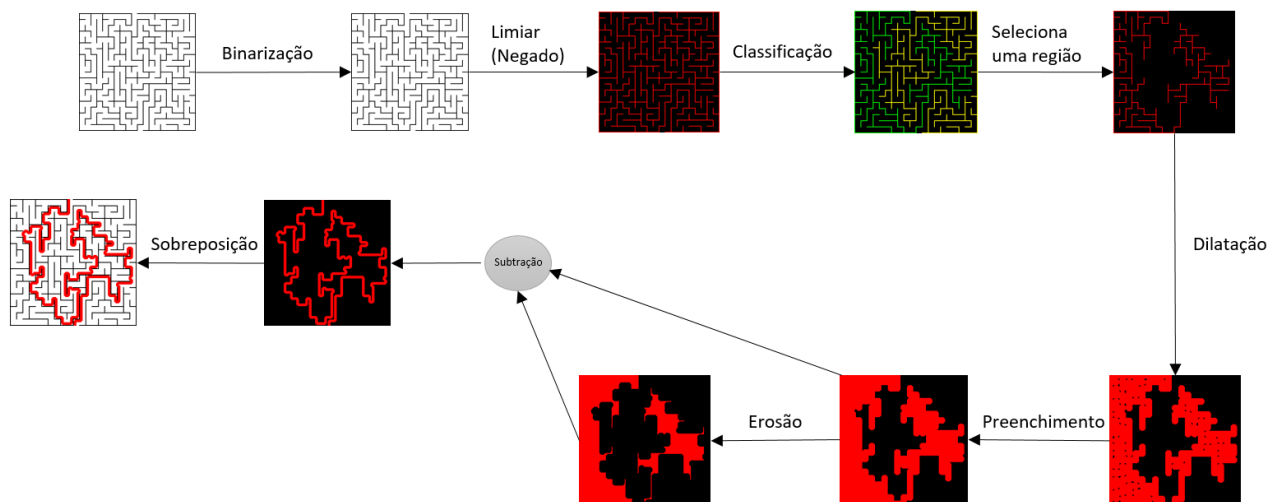


- a) Imagem após a aplicação da Esqueletização
- b) Apresentação dos pontos finais em branco
- c) retirada iterativa dos pontos finais, chegando ao caminho de saída.

2.2. SISTEMA BASEADO EM MORFOLOGIA MATEMÁTICA

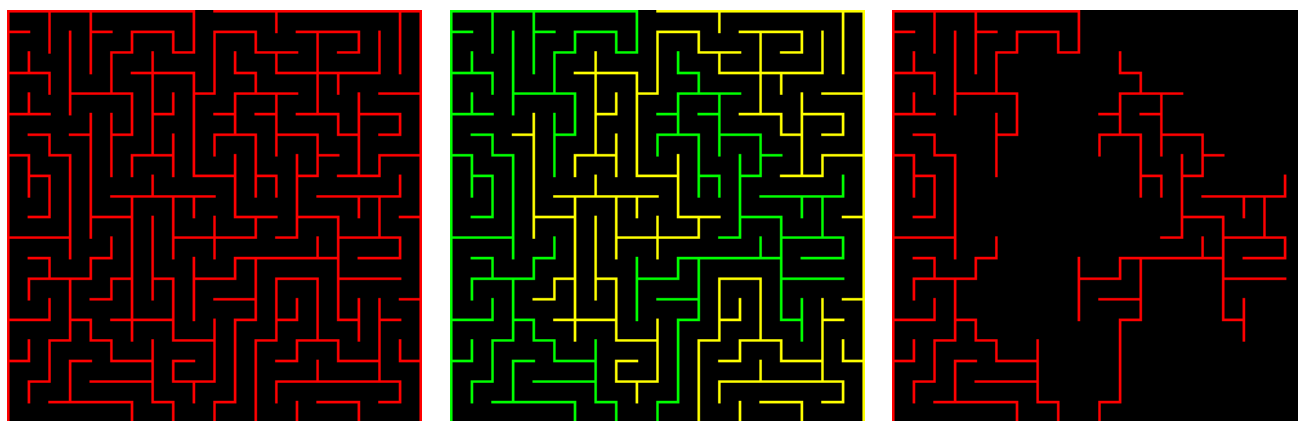
A Figura 3.5 apresenta o algoritmo geral do processamento baseado em operações básicas da Morfologia Matemática, exibindo os resultados das etapas principais executadas pelo método estudado.

Figura 3.5. Arquitetura do processamento utilizando Morfologia Matemática. Fonte: Elaborada pelo autor



A proposta inicia o processamento transformando o labirinto em uma imagem binária, e depois esses valores são invertidos através da operação de negação, que é apresentada na Figura 3.6a, em que utilizou-se a cor vermelha apenas para facilitar a visualização.

Figura 3.6. Imagens intermediárias do processo via Morfologia Matemática. Fonte: Elaborada pelo autor.

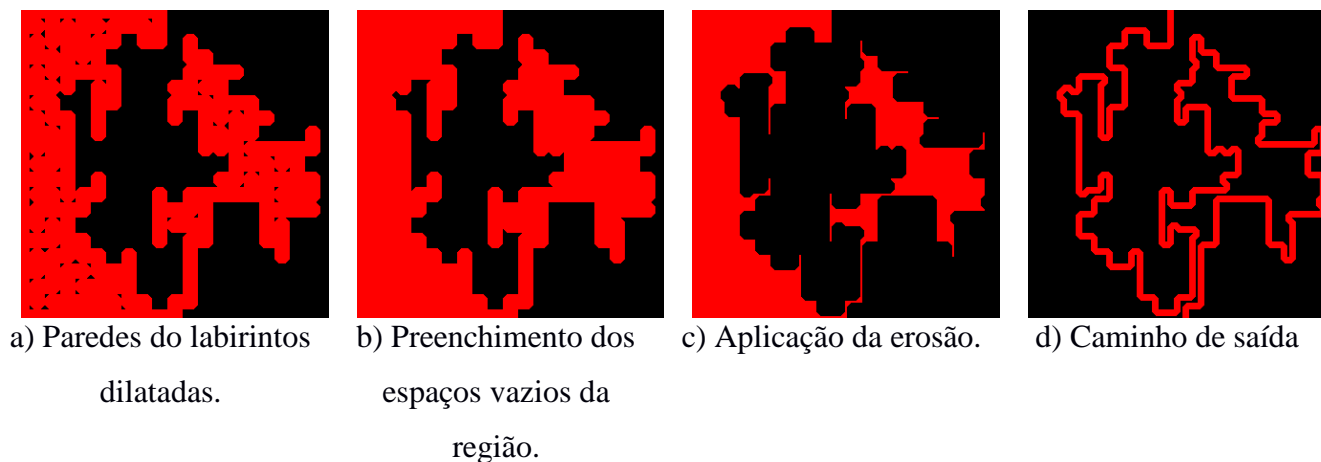


- a) Imagem binária negada, onde os pontos vermelhos têm valor 1.
b) imagem com os elementos conexos de a)
c) Seleciona-se a primeira região conexa encontrada.

No processo “Classificação” foi aplicada a verificação de conectividade-8. O resultado gerado da chamada da função pode ser visualizado na Figura 3.6b, onde é possível verificar que há duas cores, verde e amarelo. Pixels da mesma cor significam que estão conectados. A rota de saída é o divisor entre as duas paredes de cores distintas. Seleciona-se uma região com a parede que faz parte da rota de saída, no caso a primeira região conexa – Figura 3.5c, e ignora-se o resto da imagem.

Aplica-se a operação de dilatação, e todas as paredes do labirinto alargam-se, como pode ser observado na Figura 3.6a. O elemento estruturante usado é circular de diâmetro 7 (sete) e foi escolhido após diversos testes utilizando diâmetros variados, menores e maiores que o escolhido, para a obtenção dos resultados. Depois, é aplicada uma função de preenchimento dos "buracos", que é uma operação de dilatação seguida de uma operação de erosão, e o resultado pode ser visualizado na Figura 3.6b. Nesse ponto, o caminho de saída está na borda interna da região vermelha.

Figura 3.6. Imagens intermediárias do processo via Morfologia Matemática. Fonte: Elaborada pelo autor.



Aplica-se a erosão na Figura 3.6b com o mesmo elemento estruturante na dilatação, assim os locais dessa borda que sofreram dilatações das paredes originais serão reduzidos próximo ao seu tamanho original. A imagem resultante da erosão é mostrada na Figura 3.6c. Subtraindo a imagem dilatada (Figura 3.6b) da imagem erodida (Figura 3.6c), isola-se a rota de saída do labirinto, obtendo-se a Figura 3.6d. Da mesma forma que as abordagens anteriores, sobrepõe-se a rota de saída sobre a imagem do labirinto original.

2.3. EXPERIMENTOS

Esta seção apresenta os experimentos realizados, que foram divididos da seguinte forma:

- o primeiro caso de testes é o mais simples em termos de complexidade, no qual se realiza o processamento em "labirintos perfeitos", ou seja, labirintos com apenas uma solução, sem seções isoladas ou loops (PULLEN, 2015). Foram criados três grupos principais de labirintos para testes, contendo imagens de 20x20, 50x50 e 100x100 células. Cada um desses grupos de imagens possui vinte labirintos que serão utilizados para testes.
- o segundo caso apresenta os resultados dos mesmos algoritmos aplicados em labirintos que possuem mais de uma rota como solução possível, aumentando o nível de complexidade dos

experimentos, dado que é necessário que a mesma solução seja capaz de encontrar duas rotas com a mesma imagem de entrada. Essas imagens foram geradas através da edição das imagens dos Labirintos Perfeitos, retirando-se uma célula de barreira, para que assim fossem gerados dois caminhos até a saída. Isso resulta em labirintos que possuem *loops*, o que poderia incrementar bastante a complexidade, principalmente nos algoritmos que funcionam através de separação de regiões. No total, foram dez imagens de testes com 20x20 células.

- o terceiro caso de testes utiliza uma imagem de alto grau de complexidade, retirada do artigo de McLendon (MCLENDON, 2001), o qual define a complexidade de resolução de labirintos e apresenta imagens diversificadas de diferentes níveis de complexidade.

O critério principal para avaliação dos métodos aqui apresentados é a qualidade do resultado do processamento da imagem, ou seja, é analisar de forma qualitativa o caminho apresentado após o processamento do labirinto. A ferramenta utilizada para a geração dos labirintos foi o Maze Generator⁴.

3. RESULTADOS E DISCUSSÃO

Nesta seção apresentam-se os resultados de cada um dos três casos de testes citados na seção 3.3.

3.1. LABIRINTOS PERFEITOS

Neste caso de testes, foram utilizadas 20 (vinte) imagens de cada um dos três tamanhos de labirintos, totalizando 60 (sessenta) imagens de testes. As Figuras 4.1, 4.2 e 4.3 apresentam um dos resultados de cada sistema em cada um dos tamanhos de labirintos.

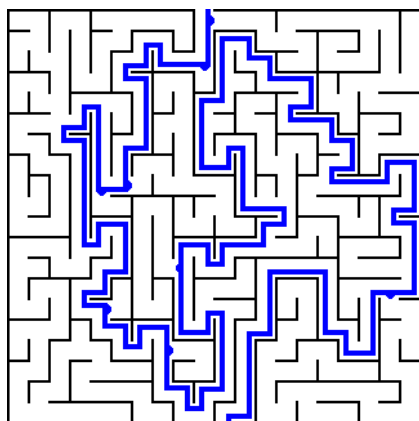
O primeiro cenário criado obteve resultados expressivos em todos os algoritmos, conforme mostrado na Tabela 4.1. O Watershed e a Morfologia Matemática atingiram uma taxa de 100% (cem por cento) de acerto, em todos os tamanhos de labirinto.

4 disponível no sítio www.mazegenerator.net

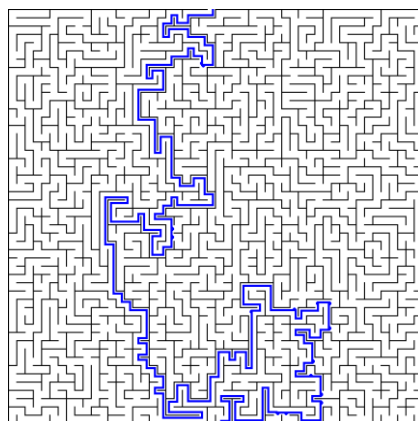
Tabela 4.1 – Comparação entre as técnicas em imagens de labirintos com apenas uma rota de saída.

Tamanho/Algoritmo	Watershed	Esqueletização	Morfologia Matemática
20 células	20/20	18/20	20/20
50 células	20/20	9/20	20/20
100 células	20/20	4/20	20/20

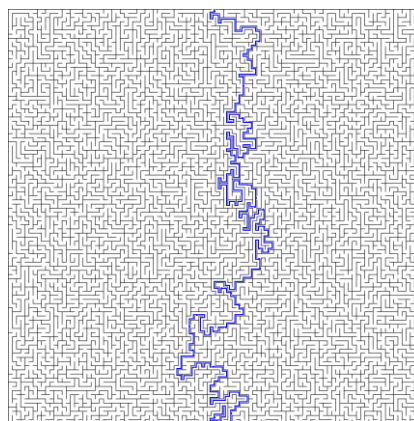
Figura 4.1. Exemplos de resultados da solução baseada em Watershed. Fonte: Elaborada pelo autor.



a) resultado do Labirinto #1 de 20x20

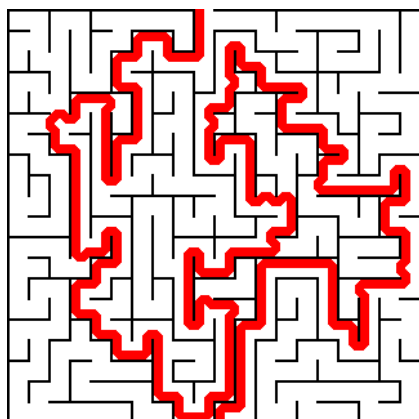


b) resultado do Labirinto #1 de 50x50

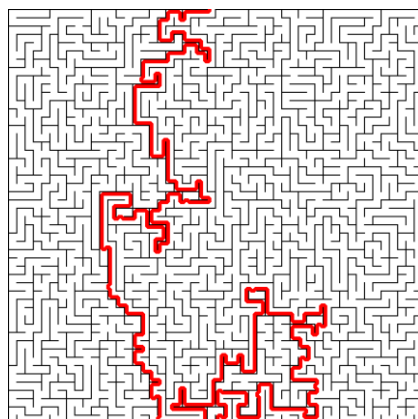


c) resultado do Labirinto #1 de 100x100

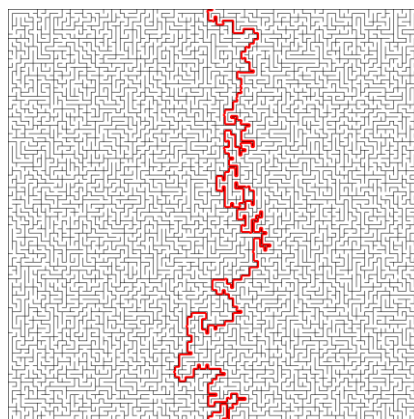
Figura 4.2. Exemplos de resultados da solução baseada em Morfologia Matemática. Fonte: Elaborada pelo autor.



a) resultado do Labirinto #1 de 20x20

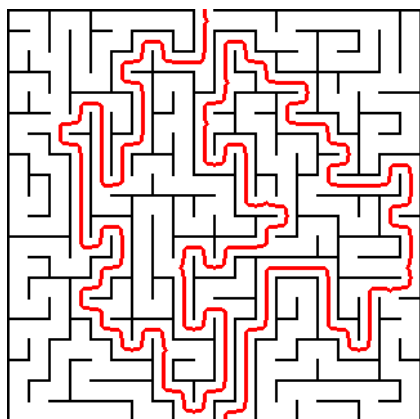


b) resultado do Labirinto #1 de 50x50

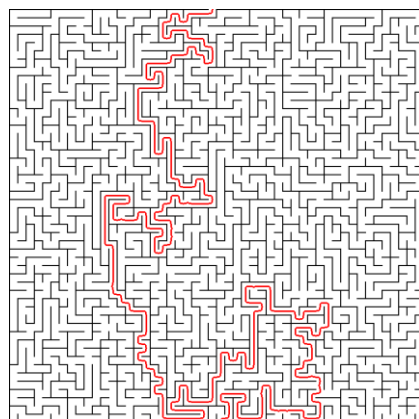


c) resultado do Labirinto #1 de 100x100

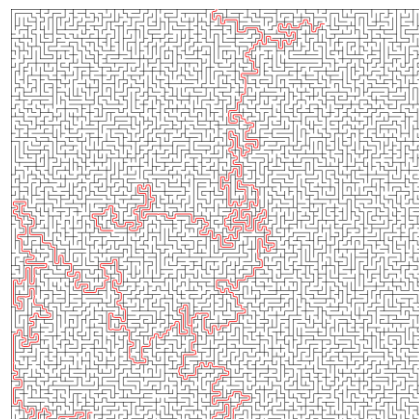
Figura 4.3. Exemplos de resultados da solução baseada em Esqueletização. Fonte: Elaborada pelo autor.



a) resultado do Labirinto #1 de 20x20

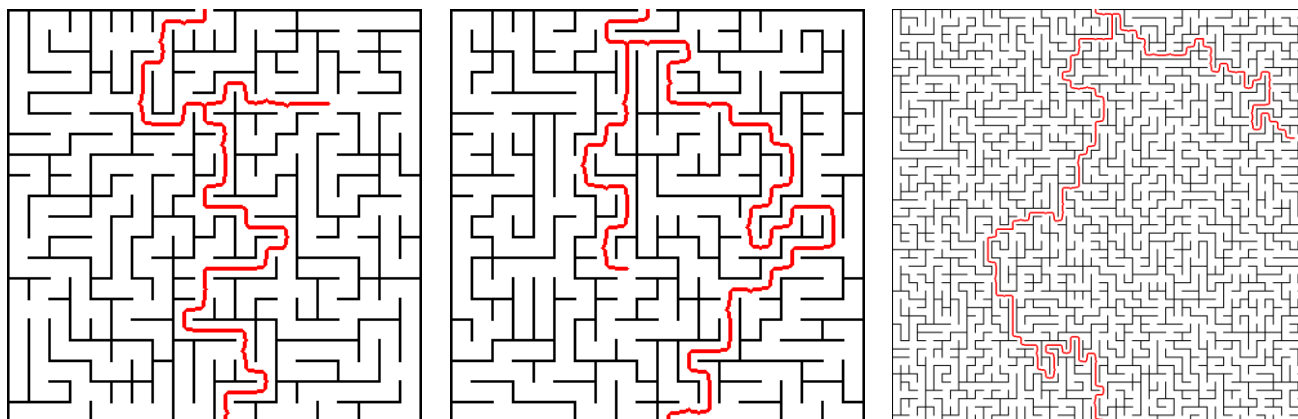


b) resultado do Labirinto #1 de 50x50



c) resultado do Labirinto #1 de 100x100

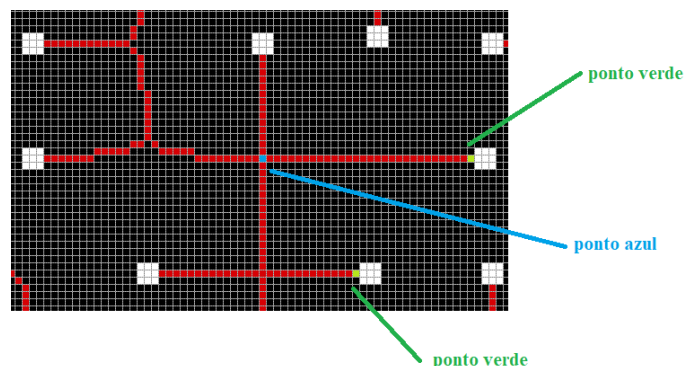
Figura 4.4. Exemplos de resultados ruins da solução baseada em Esqueletização. Fonte: Elaborada pelo autor.



A Esqueletização apresentou resultados bem inferiores. Esses resultados tiveram falhas no caminho gerado, ou seja, apesar de o processamento obter a rota de saída, informações errôneas (ramificações) foram obtidas em alguns casos, tal como pode ser visto na Figura 4.3c e nos exemplos da Figura 4.4. Portanto, é possível observar que as imagens de 20 (vinte) células obtiveram uma taxa de sucesso de 90% de acerto, as de 50 (cinquenta) células 45%, e as de 100 (cem) células 20%.

Supõe-se que a causa deste problema se deve ao algoritmo de remoção dos pontos finais do esqueleto gerado. As ramificações extras foram observadas em pontos onde existem encruzilhadas de caminhos. Essas encruzilhadas possuem pelo menos dois pontos finais com distância equivalente até o ponto central da encruzilhada, ou seja, com o mesmo número de pixels. Um exemplo é mostrado na Figura 4.5, que é uma amplificação do ponto final da falha da Figura 4.4b, onde os pontos finais são os quadrados brancos. A partir dos pontos finais, marcados em verde, ao chegar no ponto da encruzilhada (destacado em azul na Figura 4.5), há o mesmo número de pixels e o algoritmo não os retira, deixando ramificações conhecidas como *Spurious Branches*.

Figura 4.5 – Exemplo de ponto de falha no algoritmo de Esqueletização . Os pontos verdes possuem a mesma distância do ponto azul (em pixels) na imagem. Fonte: Elaborada pelo autor.

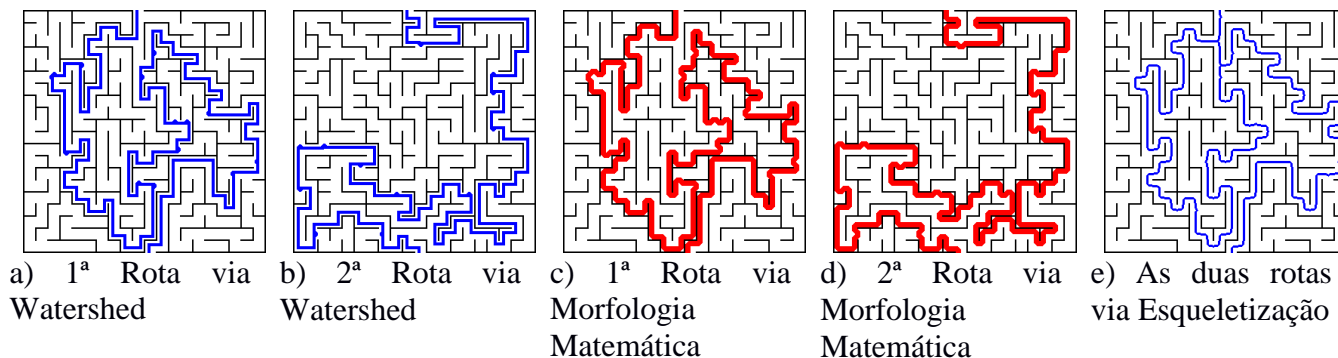


LABIRINTOS COM DUAS ROTAS DE SAÍDA

O segundo caso de testes apresenta os resultados dos mesmos algoritmos aplicados em labirintos que possuem apenas uma rota como solução possível para labirintos com duas rotas de saída. Assim, viabilizando a análise de sua aplicabilidade dos métodos em cenários mais complexos. É importante ressaltar que, devido ao fato de dois dos três algoritmos funcionarem através da separação de regiões (Watershed e Morfologia Matemática), foi necessária a execução desses algoritmos duas vezes, obtendo duas diferentes rotas de saída.

A Figura 4.6 apresenta os resultados para o teste de caso #1, onde a) e b) são as duas rotas via abordagem Watershed, c) e d) são as duas rotas via abordagem Morfologia Matemática e e) é a imagem resultante via abordagem Esqueletização, já com as duas rotas. É interessante comparar que as duas rotas dadas pelas abordagens de Watershed e Morfologia Matemática são as mesmas, sendo que o resultado via Morfologia apresenta uma primeira rota igual e outra segunda rota diferente das anteriores. Via Morfologia Matemática há dois pontos de encruzilhada que podem ser analisados como diferentes rotas: caso se siga o percurso à esquerda, na segunda encruzilhada, tem-se uma rota menor que a primeira rota via Watershed e Morfologia Matemática.

Figura 4.6. Exemplos de resultados para o teste de caso #1 de labirinto com duas rotas de saída. Fonte: Elaborada pelo autor.



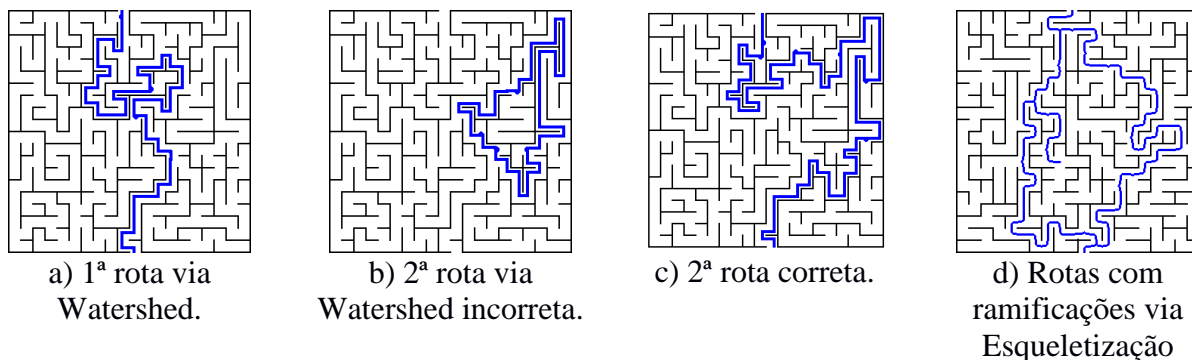
Os resultados dos testes são resumidos na Tabela 4.2. Pela abordagem via Morfologia Matemática, a taxa de acerto foi de 100%; por Watershed, de 95%; e por Esqueletização, de 90%.

Tabela 4.2 – Comparação entre as técnicas em imagens de labirintos com duas rotas de saída. Fonte: Elaborada pelo autor.

Tamanho/Algoritmo	Watershed	Esqueletização	Morfologia
20 células	19/20	18/20	20/20

Alguns dos resultados incorretos são apresentados na Figura 4.7. Em a) e em b) são as rotas via abordagem Watershed, em que a segunda rota é um caminho que não tem conexão com a entrada ou a saída do labirinto. A segunda rota correta deveria ser a apresentada na Figura 4.7c. E na Figura 4.7d apresenta-se o resultado da abordagem via Esqueletização, que mostra ramificações desconexas.

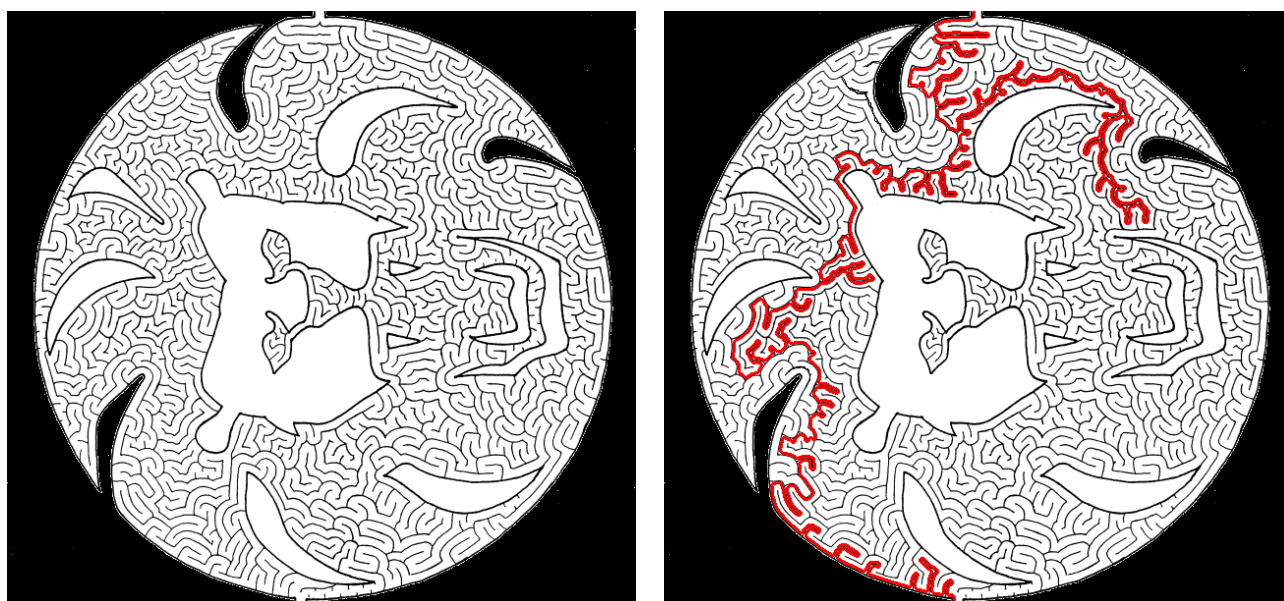
Figura 4.7. Resultados incorretos de labirintos com duas rotas de saída. Fonte: Elaborada pelo autor.



3.2. CASOS ESPECIAIS DE LABIRINTOS

Com o objetivo de explorar os limites das técnicas utilizadas, os algoritmos foram testados em labirintos de alta complexidade. Utilizou-se um dos labirintos do artigo de McClendon (2001) (Figura 4.8a), cuja forma difere dos labirintos estudados até o momento. Dentre todas as técnicas, a única que apresentou um caminho de saída desse labirinto foi a Morfologia Matemática (Figura 4.8b), todas as outras apresentaram resultados que não geravam rota de saída plausível.

Figura 4.8. Labirinto complexo e o resultado via Morfologia Matemática.



a) imagem de entrada, adaptada de (MCCLENDON, 2001).

b) resultado da abordagem via Morfologia Matemática.

4. CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi o de comparar 3 diferentes técnicas de PDI para solucionar o problema de labirintos, sendo eles *Watershed*, Esqueletização e Morfologia Matemática. Ao final da comparação qualitativa, analisou-se que a abordagem via Morfologia Matemática obteve o melhor resultado entre as técnicas, em todos os cenários de testes. Através da separação em regiões, de forma a gerar melhores resultados, a técnica teve 100% de acertos para Labirintos Perfeitos e com duas rotas de saída, e foi a única a obter sucesso pela mistura de duas ou mais regiões. Portanto, podemos afirmar que a Morfologia obteve os melhores resultados para o estudo. Também foi possível avaliar que o pior resultado foi a abordagem via Esqueletização, que gera muitas ramificações em suas rotas.

Um dos trabalhos futuros é incluir a avaliação de validade da rota de saída, além de desenvolver uma técnica para detectar o menor caminho possível quando há mais de uma rota de saída. Isso é possível incluindo-se, ao final, a utilização de grafos e algoritmos como os de Dijkstra.

5. REFERÊNCIAS

ALSUBAIE, M. **Algorithms for maze solving robot**. Faculty of Science and Engineering, Manchester Metropolitan University, 2013.

AWHAD, G. et al. **Macromouse: A maze solving robot final report**. University of Minnesota Minneapolis: Department of Computer Science and Engineering, MN, v. 55455, 2010.

CHAUDHARY, P.; GARG, G. Fungus infestation and survival a cellular automata based approach to solving digital maze images. **International Journal of Scientific and Research Publications**, v. 3, n. 6, p. 710, 2013.

COUPRIE, M.; BERTRAND, G. **Topological grayscale watershed transformation**. In: Proceedings of SPIE Vision Geometry VI, v. 3168, p. 136–146, 1997.

ELSHAMARKA, Ibrahim, SAMAN, Abu Bakar Sayuti. Design and implementation of a robot for maze-solving using flood-fill algorithm. **International Journal of Computer Applications**, v. 56, n. 5, p. 8-13, out. 2012.

FISHER, R. et al. **Morphology - Dilation**. 2003. Disponível em: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>>.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc, 2006.

KATHE, O.; TURKAR, V.; JAGTAP, A.; GIDAYE, G. **Maze solving robot using image processing**. In: Proceedings of 2015 IEEE Bombay Section Symposium (IBSS), p. 1-5, set. 2015.

LIU, D.; SORAN, B.; PETRIE, G.; SHAPIRO, L. A review of computer vision segmentation algorithms. **Lecture Notes**, v. 53, 2012.

LUENGO, C. **Solving mazes using image analysis**. 2010. Disponível em: <<http://www.crisluengo.net/index.php/archives/277>>. Acesso em: 7 fev. 2016.

MARKOSKA, R. et al. **Cognitive mapping in robotics using genetic algorithms**. 2015. Disponível em: <https://www.researchgate.net/profile/Ramona_Markoska/publication/273693148_ICAIT13_RMarkoska/links/5508880b0cf2d7a2812a32a8.pdf>. Acesso em: 20 jan. 2016.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

MATHWORKS, I. T. **Tips and Tricks: Solving a maze with the watershed transform**. 2014. Disponível em: <<http://www.mathworks.com/company/newsletters/articles/tips-and-tricks-solving-a-maze-with-the-watershed-transform.html>>. Acesso em: 20 jan. 2016.

MATHWORKS, I. T. **Watershed transform - MATLAB Watershed**. 2016. Disponível em: <http://www.mathworks.com/help/images/ref/watershed.html?s_tid=srchtitle>. Acesso em: 16 maio 2016.

MCCLENDON, M. S. The complexity and difficulty of a maze. **BRIDGES – Mathematical Connections in Art, Music and Science**, p. 213–222, 2001.

MURATA, Y.; MITANI, Y. A fast and shorter path finding method for maze images by image processing techniques and graph theory. **Journal of Image and Graphics**, v. 2, n. 1, p. 89, 2014.

PARKER, J. R. **Algorithms for Image Processing and Computer Vision**. Wiley: IT Pro, 2010.

PLOTZE, R. O.; BRUNO, O. M. **Estudo e comparação de algoritmos de esqueletonização para imagens binárias**. IV Congresso Brasileiro de Computação - CBComp, 2004.

PULLEN, W. D. **Think Labyrinth: Maze Algorithms**. 2015. Disponível em: <<http://www.astrolog.org/labyrinth/algrithm.htm>>. Acesso em: 18 jan. 2016.

SINGH, A.; SEKHON, G. S. A new shortest path finding algorithm for a maze solving robot with simulator. **International Journal of Computer Science and Communication**, v. 2, n. 2, p. 445–449, 2011.

SINGH, B.; PANDEY, A. K. **Maze using hybrid genetic algorithm**. In: Proceedings of International Conference of Advance Research and Innovation (ICARI 2014). [S.l.: s.n.], 2014. p. 212–216.

VLIET, L. J. **DIPlib & DIPIImage**. 2003. Disponível em: <<http://www.diplib.org/>>. Acesso em: 20 abr. 2016.