Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Computational Intelligence Society, Vol. 11, Iss. 2, pp. 103-122, 2013

APLICAÇÃO DA REDE NEURAL SEM PESO WISARD PARA O RASTREAMENTO DE ALVOS DE SUPERFÍCIE NO MAR

Rodrigo da Silva Moreira
Nelson Francisco Favilla Ebecken

Universidade Federal do Rio de Janeiro
Programa de Engenharia Civil da COPPE
e-mails: moreira@ipqm.mar.mil.br; nelson@ntt.ufrj.br
Cidade Universitária, Ilha do Fundão, Centro de Tecnologia, Bloco B
CP 68506, CEP 21945-970, Rio de Janeiro, RJ, Brasil

Resumo- Neste artigo é apresentado um método de rastreamento de alvos de superfície em vídeo utilizando a rede neural sem peso *WiSARD* dado somente a informação sobre sua localização no primeiro quadro. O rastreamento de objetos em vídeo é uma tarefa importante e desafiadora em muitas aplicações. Dificuldades podem surgir devido à condição climática, à trajetória, ao aspecto do alvo, à oclusões, à condições de iluminação e à presença de ruído. O rastreamento é uma aplicação de alto nível e requer a localização do objeto quadro a quadro em tempo real. Os três principais passos executados a cada quadro por um rastreador baseado na detecção por segmentação são: a detecção, o acompanhamento e a análise das características do objeto. Estes passos dependem da qualidade da segmentação. O rastreamento realizado com a rede neural *WiSARD* depende da qualidade da binarização. Este artigo propõe uma forma rápida de binarização híbrida (limiarização e detecção de borda) no modelo de cor *YCbCr* e maneiras de configurar uma rede neural *WiSARD* para melhorar a eficiência do rastreador quando erros de binarização ocorrem.

Palavras-chave- Redes neurais sem peso, rede neural *WiSARD*, segmentação de imagens, binarização de *pixels*, modelos de cor, rastreamento de objetos, alvos de superfície.

1 Introdução

O rastreamento de alvos de superfície (todas as embarcações, navios ou barcos inimigos) é uma tarefa de grande importância para navios de guerra da Marinha. O grande avanço da eletrônica e o surgimento de novas tecnologias possibilitaram o emprego de diferentes tipos de armas dentro do cenário tático. Todos os tipos de armas são apoiados por um sistema de rastreamento de alvos que utilizam sensores e equipamentos como o odômetro, o anemômetro, a giro, o radar, o sonar, a EOS (equipamento responsável por captar imagens do cenário tático) e o MAGE [1] (Medida de Apoio a Guerra Eletrônica – classificador de ondas eletromagnéticas) com a finalidade de calcular qual a melhor direção e sentido para um disparo.

O rastreamento por vídeo [2,3,4] é uma ferramenta que substitui ou auxilia os sistemas de rastreamento por radar. Muitos fatores dificultam o acompanhamento por radar [5]. Entre eles estão as condições climáticas que geram *cluster* (espécie de ruído) nos sinais de radar, a presença de zonas não cobertas pelo radar, os sistemas de contramedidas eletrônicas que geram sinais eletromagnéticos capazes de saturar ou enganar os sinais recebidos pela antena do radar e o elevado custo do *hardware* envolvido. O sistema de rastreamento de alvos por vídeo utiliza os sinais das EOS (alças optrônicas), responsáveis por captar imagens do cenário tático onde se encontra o navio para detectar o alvo de superfície e realizar o rastreamento em tempo real.

Este artigo apresenta um rastreador de alvos de superfície em vídeo baseado na rede neural sem peso *WiSARD* [6,7,8] que é eficiente, executado em tempo real e que consegue compensar erros de binarização gerados na etapa de segmentação de cada quadro. Para atingir a melhor configuração (compensar melhor os erros de binarização, aumentar a precisão e executar em tempo real) do rastreador *WiSARD*, foram testados vários tamanhos de nós *RAM* em redes neurais *WiSARD* trabalhando individualmente e em paralelo com outras redes *WiSARD,* os principais métodos de segmentação de imagens e de modelos de cor e diferentes formas de preditores de posição simples.

1.1 O processo de independência tecnologia

Um país que está em desenvolvimento e que almeja obter um maior espaço no cenário mundial, defender a Amazônia Azul, garantir a vigilância e proteção de seus interesses no mar [9], deve chegar a um estado de independência econômica, financeira e tecnológica. Para alcançar a independência tecnológica, o domínio sobre o conhecimento deve ser conquistado e somado à

atual capacidade de operar e manter equipamentos. Para obter a conquista, devem existir incentivos no campo de pesquisa e do desenvolvimento, aumentando, consequentemente, a capacidade de criação de novos engenhos, de baixo custo, eficazes e capazes de realizar tarefas que antes somente eram conseguidas através da importação.

1.2 Rastreadores de objetos

O rastreamento de objetos é em geral um problema desafiador [10] e já foi bastante estudado. São empregados em sistemas de vigilância [11], onde atividades consideradas suspeitas geram um alerta [12]. Também são empregados para rastrear alvos por um sistema de armas, reconhecer gestos e movimentos humanos, auxiliar a condução de carros, monitorar o tráfego nas ruas, auxiliar um míssil a atingir seu alvo ou um robô a se movimentar entre obstáculos por navegação inercial. Foram desenvolvidos rastreadores baseados na remoção do cenário de fundo [13], *Global Nearest Neightbor* [14], pontos característicos [15], fusão de áudio e vídeo [16], *distribution fields (DF)* [17], modelos esparsos de aparência [18], modelos de Markov [19], modelos estatísticos [20], filtro de partículas [2], transformadas *Wavelet* [3], diferença entre quadros consecutivos [4], s*upport vector machine* [21]*,* redes neurais artificiais [22], *feed-forward* [23] e *Fuzzy* [24]. Rastreadores baseados em redes neurais sem peso não estão difundidos. Este artigo propõe um rastreador de alvos de superfície baseado na rede neural sem peso *WiSARD*.

Os sistemas de rastreamento possuem em geral três processos principais [10]: detecção do objeto, acompanhamento quadro a quadro e análise do objeto e seu comportamento. As principais categorias de detecção de objetos são: detecção por pontos, segmentadores, modeladores do cenário de fundo e classificadores supervisionados. Questões como qual representação de objeto é mais adequada, quais características (*features*) do objeto devem ser consideradas e como melhor representar sua aparência e movimento devem ser estudadas para cada aplicação. As formas de representação de objetos mais conhecidas são: por pontos, por formas geométricas primitivas, pelo contorno, pelo esqueleto e por corpos articulados. A aparência pode ser modelada utilizando funções densidade de probabilidade, *templates*, modelos ativos (forma e aparência são modeladas juntas) e múltiplas vistas. Aspectos como cor, bordas, pontos importantes, fluxo ótico e textura podem ser aproveitados para o rastreamento. A análise da trajetória pode ser realizada, entre outras formas, com filtros de Kalman, *Meanshift*, *KLT*, *Block-matching*, preditores de segunda e primeira ordem e redes neurais artificiais.

A complexidade do rastreamento de alvos de superfície em vídeo advém de diversos fatores como: as condições marítimas e climáticas; a velocidade do seu navio; a velocidade, o tamanho, a proximidade e aparência do alvo; e a elevada quantidade de embarcações não hostis. Dependendo do estado do mar, o alvo e o navio podem jogar e caturrar (movimentos ao longo dos eixo longitudinal e transversal do navio respectivamente) dificultando o acompanhamento mesmo com a presença de câmeras estabilizadas. A presença de chuvas, ventos fortes, nebulosidade e nuvens carregadas podem alterar o nível de luminosidade e com isso o contraste de cores diminui. Dependendo do tamanho do alvo no vídeo e de sua manobra evasiva, o algoritmo de rastreamento pode perdê-lo. A presença de outras embarcações na área de patrulhamento pode confundir ou obstruir a vista do alvo. A aparência pode variar devido à mudança de sua orientação, às imperfeições na câmera, à luminosidade e oclusões parciais ou totais. O algoritmo deve ser capaz de rastrear o alvo mesmo nestas circunstâncias.

Rastreadores que utilizam a metodologia de rastreamento por detecção (*tracking-by-detection*) e cujos detectores são baseados em segmentação de imagens (detectores segmentadores) obtêm as informações do alvo após tentar separá-lo do cenário de fundo por segmentação. Esta nunca é perfeita, portanto no detector existe um ou mais classificadores responsáveis por retornar a parte de cada quadro do vídeo que mais se aproxima do modelo do alvo. Classificadores baseados em redes neurais artificiais, *SVM* e redes bayesianas [10] são bastante empregados para esta tarefa. O alvo é procurado apenas em uma área denominada janela de busca (pequena região de cada quadro) para diminuir o tempo de execução. Um preditor de posição pode auxiliar o posicionamento do centro da janela de busca no quadro seguinte.

O rastreador apresentado neste artigo acompanha alvos em vídeo utilizando o método de rastreamento por detecção. Possui dois módulos principais: o detector e o preditor de posição. O detector é responsável por retornar a posição do alvo em cada quadro. O preditor analisa as posições do alvo nos últimos 3 quadros para indicar a provável posição do alvo no próximo quadro. O detector é baseado em segmentação de imagens e utiliza um classificador baseado na rede neural sem peso *WiSARD* (seção 2). É composto por um conjunto de discriminadores deslocados espacialmente uns em relação aos outros. Cada discriminador possui um conjunto de memórias *RAM* (nós da rede *WiSARD* – figura 2) que são endereçadas por um conjunto de *bits* (entradas do nó) resultantes da binarização de *pixels*. No primeiro quadro, os discriminadores da rede são treinados com o padrão resultante da binarização dos *pixels* da janela de seleção definida manualmente pelo operador (este padrão de *bits* é o modelo do alvo a ser seguido). Nos quadros seguintes, a rede recebe como entrada o resultado da segmentação (binarização) dos *pixels* da janela de busca de cada quadro. Cada discriminador fica responsável por uma porcentagem da área de busca (figura 3) e compara o modelo treinado com os *pixels* binarizados da área de sua responsabilidade, retornando um número

inteiro. Quanto maior for este valor, mais o modelo é similar aos *pixels* binarizados. O discriminador que retorna a maior resposta revela a posição do alvo (seção 2.1). O rastreador foi desenvolvido em *C/C++* e sua interface homem-máquina (IHM – figura 1) foi implementada em *QT*. Os vídeos são executados em tempo real em um *Laptop HP* com 298MB de memória *RAM*, 1.14GB de *HD*, processador *AMD Turion(tm)* 64 *Mobile* ML-30 e sistema operacional *Fedora release* 11 com *Genome* versão 2.26.1 e *kernel linux* 2.6.29.4-167.fc11.i686.PAE. Propositadamente, foi utilizado um computador com pouca capacidade de processamento e sistema operacional desatualizado para testar se esta abordagem de rastreamento consegue acompanhar o alvo em tempo real, mesmo sem uma configuração de *hardware* e sistema operacional de última geração.

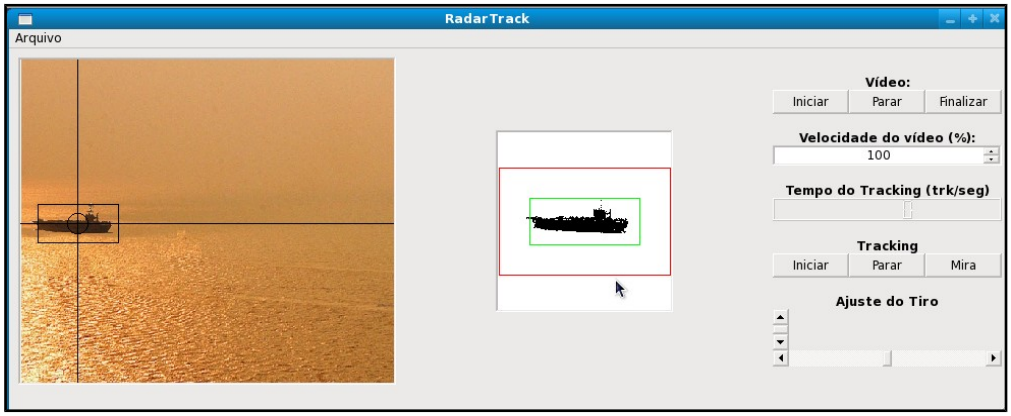


Figura 1: Interface do rastreador baseado na rede neural sem peso *WiSARD*

2 A rede neural sem peso *WiSARD*

As primeiras pesquisas na área de redes neurais artificiais surgiram na década de 40 com os precursores Warren McCulloch e Walter Pitts. Eles buscaram desenvolver um modelo matemático de um neurônio a partir de pesquisas do neurônio natural do cérebro humano [25]. Incentivados pelo estudo iniciado por McCulloch e Pitts, pesquisadores da área buscaram desenvolver, através da união de múltiplos neurônios, redes capazes de aprender e reconhecer padrões fornecidos como entrada.

Na área de redes neurais artificiais uma das vertentes principais de pesquisa é a de redes neurais sem pesos. Nestas redes, as entradas e a saída dos neurônios são números binários (0 ou 1) e não existem pesos entre os neurônios. As funções de cada neurônio são armazenadas em *look-up tables* que podem ser implementadas com memórias *RAM* [26]. O aprendizado das redes neurais com pesos envolve o ajuste de pesos. Diferentemente destas redes, o processo de aprendizagem das redes neurais sem pesos é realizado através da modificação das palavras armazenadas nas *look-up tables*, permitindo a construção de algoritmos flexíveis e o aprendizado rápido. Com a utilização de *look-up tables*, qualquer função pode ser implementada nos nós já que qualquer valor binário pode ser armazenado em resposta a um conjunto de *bits* de entrada durante o treinamento. A rapidez do aprendizado deve-se à independência mútua entre os nós quando os dados de entrada da rede são modificados. A alteração do valor dos pesos durante o aprendizado de um mapeamento de entrada e saída nas redes neurais com pesos muda o comportamento do nó em relação a padrões previamente treinados.

A rede neural *WiSARD* foi criada por Wilkes, Stonham e Aleksander em 1984 [27]. É uma rede neural sem pesos [6] onde os neurônios são implementados com memórias *RAM*. Esta forma de implementação consegue contornar o problema da impossibilidade de implementação da função *Exclusive OR* pelo modelo de neurônio de McCulloch and Pitts – o *perceptron*, além de apresentar a vantagem de ser treinada em um tempo muito curto, essencial para a aplicação abordada no presente artigo. *WiSARD* é uma rede neural destinada para o reconhecimento de padrões mas também pode ser utilizada com outros fins, como por exemplo o rastreamento de alvos (este artigo), o controle automático dos movimentos de uma plataforma *offshore* [7], o reconhecimento de padrões e a construção de modelos mais representativos para padrões previamente treinados [28], a minimização do problema de saturação dos neurônios [29], a classificação de portais [30], o desenvolvimento de sistemas de vigilância [8], a localização de um robô durante sua navegação em uma sala de escritório [31] e o diagnóstico de distúrbios neuromusculares [32]. A rede *WiSARD* pode ser implementada em *hardware* [33]. Alguns artigos e livros descrevem o princípio de funcionamento da rede neural *WiSARD* [6,8].

[7] propôs a criação de um sistema visual de controle automático dos movimentos de uma plataforma *offshore* utilizando apenas câmeras como sensores para auxiliar nas operações de carga e descarga em um navio no mar. O controle depende da modelagem do movimento do navio (seis eixos de liberdade). Para controlar os movimentos da plataforma, uma rede *WiSARD*

identifica e segue alguns pontos de referência do convés do navio nas imagens binarizadas obtidas por cada câmera. Com os pontos de referência, o movimento é modelado e as operações de carga e descarga podem ser realizadas.

A rede *WiSARD* pode ser utilizada para o reconhecimento de padrões [28]. Contando a frequência com que os endereços de cada nó *RAM* são acessados durante a fase de treinamento, é possível associar os endereços mais acessados com os padrões utilizados na fase de treinamento. Desta forma, modelos mais representativos de cada classe são criados (*DRASiW*).

Um sistema de classificação de portais encontrados em construções antigas na Itália utilizando a rede *WiSARD* foi proposto por [30]. Quando portais têm aspectos semelhantes, a classificação é desafiadora e muitos erros podem ocorrer. O sistema possui dois módulos: um reconhece os aspectos geométricos de partes do portal com a rede *WiSARD* e o outro integra as informações geométricas obtidas por esta rede para classificar o portal através de testes sucessivos de hipóteses.

Para diminuir o problema de saturação dos neurônios, [29] propôs uma modificação do algoritmo de treinamento original da *WiSARD*. Uma medida que combina os conceitos de validação cruzada e da teoria da informação de Shannon é utilizada durante a fase de treinamento para a seleção das melhores conexões e consequentemente atingir uma perfórmance melhor.

Na implementação do sistema de vigilância utilizando a rede neural *WiSARD* proposta por [8], apenas as partes mais relevantes dos quadros de um vídeo (locais onde não pode haver movimento de pessoas) são binarizados. Os *pixels* binarizados são colocados na entrada do módulo *WiSARD* do sistema. Quando a ocorrência de uma anormalidade é detectada, como por exemplo, de pessoas se movimentando sobre trilhos de trem, um alerta é gerado.

A rede *WiSARD* agindo em conjunto com um módulo *NSP* (*Neuro Symbolic Processor*) pode ser utilizada para localizar um robô dentro de um ambiente fechado (*office like enviroment*) que não possui degraus ou escadas [31]. Um mapa planar 2D contendo todas as informações dos cantos da sala é passado para o robô por uma câmera digital. A extração dos cantos contidos no mapa é feita por uma rede *WiSARD*. Durante a navegação do robô, outra rede *WiSARD* pega os quadros do vídeo captado por uma câmera presa ao robô para detectar os cantos presentes no local. Estes cantos são passados para o módulo *NSP* que tenta decidir qual lugar no mapa o robô se encontra relacionando os cantos detectados com as informações do mapa.

Três tipos de redes foram comparadas com a rede *WiSARD* na performance da diagnose de desordens neuromusculares em [32]. Apesar de a rede *WiSARD* obter uma performance semelhante às das outras redes, o tempo de treinamento obtido foi muito menor. Esta característica da rede *WiSARD* é muito importante para as aplicações de rastreamento de alvos.

[33] apresenta uma forma de implementação em *hardware* da rede *WiSARD* utilizando *Pyton* e *VHDL*. Por se tratar de uma rede onde os nós são memórias *RAM*, a implementação em hardware se torna bastante simples.

A rede neural *WiSARD* é construída agrupando um conjunto de elementos básicos denominados discriminadores. Cada discriminador é responsável por reconhecer uma classe diferente de padrões. Os discriminadores são compostos por nós que são memórias *RAM*. Cada memória *RAM* armazena palavras de 1 *bit* (0 ou 1). Portando, quando no barramento de endereços de um dos nós de um discriminador é colocado um conjunto *C1* de *bits*, o nó gera na sua saída o *bit* previamente armazenado no endereço acessado pelo conjunto *C1* durante a fase de treinamento. A resposta de um discriminador é um número inteiro formado pela soma dos *bits* gerados nas saídas de todos os seus nós. Quanto maior for a resposta do discriminador, maior é a probabilidade do dado de entrada do discriminador pertencer à classe que ele representa. Dado um conjunto de entrada $C = \{C1,C2,...,CK\}$, onde cada elemento de *C* é um vetor de *bits* de mesmo tamanho. *C1* é colocado no barramento de endereços do primeiro nó de todos os discriminadores da rede *WiSARD*. *C2* é colocado no barramento de endereços do segundo nó de todos os discriminadores da rede *WiSARD*. E assim é feito sucessivamente até chegar ao último vetor *CK*. Todos os discriminadores da rede irão retornar um valor inteiro. No caso mais simples de decisão, o conjunto de entrada *C* pertence à classe representada pelo discriminador que gerou a maior resposta.

A rede neural *WiSARD* pode ser definida como uma rede composta por um conjunto de discriminadores, onde cada discriminador é formado por um conjunto de *k* nós *RAM* (figura 2), cada um endereçado por *N bits*. Cada uma das *k* memórias *RAM* armazena palavras de um *bit*, num total de 2^N *bits* [26]. O conjunto de *N bits* de endereçamento é escolhido aleatoriamente de um vetor de entrada *E* contendo *k*N bits*. No contexto de rastreamento por vídeo, as imagens de cada quadro de tamanho *k*N pixels* são binarizadas (cada valor *RGB* de um *pixel* é transformado em um *bit*). Uma imagem binarizada é uma matriz de dimensão *W* x *H*, mas pode ser interpretada como um vetor de dimensão *k*N bits (W*H = k*N)* se cada linha da matriz for colocada uma ao lado da outra para montar o vetor. No caso de uma imagem contendo *k*N pixels*, cada *pixel* binarizado representa um único *bit*. O barramento de endereços de cada nó *RAM* é conectado à *N bit*s. *k* conjuntos de *N* posições (*bits*) escolhidos aleatoriamente do vetor *E* são conectadas ao barramento de endereços dos *k* nós *RAM* de um

discriminador. Cada nó *RAM* tem um dos seus endereços acessados. Mas em cada discriminador da rede *WiSARD,* os nós *RAM* são acessados em paralelo, totalizando *k* palavras de 1 *bit* acessadas simultaneamente quando uma imagem binarizada é colocada na entrada da rede. Cada discriminador forma sua saída por um somador que realiza a soma sem pesos das *k* palavras acessadas pelas *k RAM* (*k bits* acessados). A saída do somador é denominada resposta do discriminador. A ligação aleatória dos *k*N bits* do vetor *E* nos *k* barramentos de endereços dos nós *RAM* é denominado mapeamento de entrada. Uma vez definido o mapeamento de entrada, este permanece constante durante a fase de treinamento e de classificação dos padrões. No problema de reconhecimento de letras manuscritas, um discriminador pode ser treinado com imagens digitalizadas e binarizadas contendo a letra A. Outro discriminador é treinado com imagens contendo a letra B e assim por diante. Colocando na entrada de cada discriminador uma mesma imagem de teste contendo uma letra a ser reconhecida, as respostas dos discriminadores serão colocadas nas suas saídas. O discriminador que retornar a maior resposta revela a letra contida na imagem de teste.

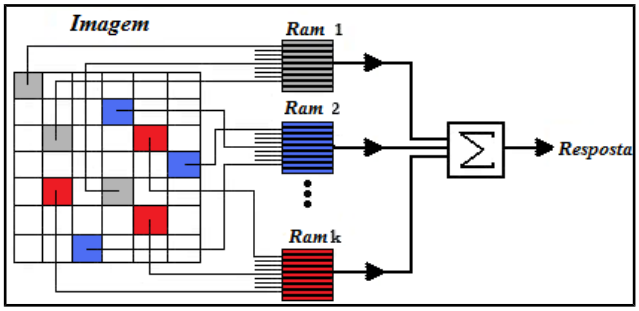


Figura 2: Discriminador *WiSARD*: A figura mostra um discriminador com *k* nós *RAM* de *9 bits* de endereçamento onde cada *bit* está conectado a um *pixel* binarizado. Estão sendo representados somente *3* dos *9 pixels* ligados ao barramento de endereços de cada *RAM*.

Possuindo mais de um nó *RAM*, os discriminadores da rede *WiSARD* conseguem compensar o problema da incapacidade de generalização que cada *RAM* sozinha apresenta [7]. Antes da fase de treinamento, todos os endereços das memórias *RAM* de todos os discriminadores são gravados com "*0*"s. Para a fase de treinamento, um conjunto de treinamento com *X* vetores de entrada contendo *k*N bits* é utilizado. No contexto de reconhecimento de letras manuscritas, os vetores do conjunto de treinamento são formados pela binarização de imagens de tamanho *k*N bits*. Cada vetor do conjunto de treinamento é colocado um por vez na entrada do discriminador da rede *WiSARD* responsável por reconhecer a letra representada pelo vetor de entrada. Neste momento, as *k* memórias *RAM* do discriminador acessam um endereço cada. Nestes endereços é gravado o *bit* "1". Para o problema de classificação de padrões (de letras, por exemplo), cada discriminador é treinado com um conjunto de vetores de treinamento representando dados de uma mesma classe, sendo, portanto responsável por representar e reconhecer uma classe de padrões. Durante a fase de reconhecimento, um dos *Y* vetores do conjunto de teste é colocado por vez na entrada de todos os discriminadores da rede *WiSARD*. Cada vetor do conjunto de teste é classificado como membro da classe representada pelo discriminador que retornar a maior resposta (maior soma).

A proposta deste artigo é desenvolver um rastreador de alvos de superfície baseado na rede neural *WiSARD* e verificar sua performance em condições adversas (seção 2.1).

2.1 Funcionamento do rastreador *WiSARD* proposto

O rastreador baseado na rede neural *WiSARD* proposto tem por objetivo principal rastrear alvos de superfície a partir de um vídeo cujos quadros são formados por *pixels* no modelo *RGB*.

Em cada quadro do vídeo os *pixels* são binarizados e inseridos na entrada da rede *WiSARD*. Os discriminadores da rede *WiSARD* do rastreador proposto não são treinados da mesma forma como uma rede responsável pelo reconhecimento de letras manuscritas (seção 2). A rede *WiSARD* deve reconhecer a posição do alvo em cada quadro. Para esta tarefa, todos os discriminadores são treinados com a mesma imagem de entrada: os *pixels* binarizados da janela de seleção. Para reconhecer letras manuscritas, cada discriminador é responsável por reconhecer uma letra. Para isto, cada discriminador é treinado com imagens de treinamento contendo somente a letra que deve reconhecer. Um discriminador é treinado com imagens contendo a

letra A, o outro discriminador é treinado com imagens contendo a letra B e assim por diante. Mas o rastreador deve retornar a posição do alvo. Para esta tarefa, todos os discriminadores são treinados com o mesmo modelo (os *pixels* binarizados da janela de seleção). Porém, os discriminadores da rede são responsáveis por reconhecer o alvo (modelo) em uma área diferente da imagem de cada quadro (figura 3). A área coberta por cada discriminador está deslocada espacialmente das demais. Portanto, cada discriminador tentará reconhecer o alvo em uma área diferente do quadro. O discriminador que retornar a maior soma com um grau de confiança *C* maior que um valor mínimo (diferença em relação a segunda maior soma) revela a posição do alvo (figura 3).

Os centros geométricos das áreas cobertas pelos discriminadores formam uma grade de pontos *GP*. As letras *GP* aparecerão em todo o texto.

O tamanho da área em *pixels* coberta por cada discriminador é igual a área da janela de seleção. A área coberta por todos os discriminadores juntos, denominada janela de busca, é maior do que a da janela de seleção. A área coberta por um discriminador tem intercessão não vazia em relação à união das áreas cobertas pelos demais discriminadores (figura 3). As principais etapas do rastreador *WiSARD* são:

1- Selecionar o alvo (ou parte dele) no primeiro quadro do vídeo. Esta área selecionada é denominada janela de seleção e deve ser definida pelo operador [34];

2- Binarizar os *pixels* pertencentes à janela de seleção;

3- Treinar a rede *WiSARD* (todos os discriminadores) com os *pixels* binarizados da janela de seleção. Os *pixels* binarizados desta janela formam o vetor de treinamento (modelo do alvo) para todos os discriminadores da rede *WiSARD*.;

4- Rastrear o alvo (padrão treinado) nos quadros posteriores ao primeiro. Em cada quadro, os *pixels* contidos na janela de busca são binarizados. Os *pixels* binarizados contidos na área de responsabilidade de cada discriminador são inseridos em sua entrada como um vetor de teste. A área de cada quadro coberta pelo discriminador que retornar a maior resposta revela a área onde está o alvo selecionado no primeiro quadro (posição do alvo).

Após a seleção inicial do alvo (passo 1) e a binarização da janela de seleção (passo 2), é definido um padrão de *pixels* para treinar a rede *WiSARD* (passo 3). Para permitir a busca do alvo (passo 4) em cada quadro, todos os discriminadores são treinados com o mesmo padrão de *pixels* (passo 2).

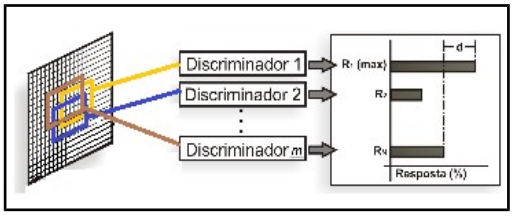


Figura 3: Distribuição espacial das áreas cobertas pelos discriminadores em um quadro do vídeo (modificado do artigo [7]).

O algoritmo do rastreador é semiautomático. A interferência humana ocorre apenas no primeiro passo. Os passos seguintes são automáticos [35]. Este rastreador pode se tornar automático trocando o primeiro passo do rastreador proposto por um módulo de vigilância capaz de detectar movimentos relevantes no vídeo, classificar (decidir) se o movimento é de um navio e retornar a janela que envolve o alvo. Para que o módulo de vigilância seja executado em tempo real, algoritmos rápidos de detecção de movimentos como o *background-subtraction* [13] e a diferença entre quadros sucessivos [15] devem ser pesquisados.

Como proposta para um aperfeiçoamento futuro do rastreador, assim que este perceber através do cálculo do grau de confiança que as respostas de dois discriminadores estão próximas, a rede deve ser retreinada com um novo modelo *online* e a configuração da rede neural deve ser mudada (seção 4.4.5). A seção 4.4.5 descreve uma forma inovadora de modificar a configuração de cada discriminador para compensar erros de binarização gerados nos passos 2 e 4.

3 Segmentação de Imagens

A segmentação é uma das mais difíceis e importantes áreas de estudo do processamento de imagens [36]. É definida como sendo o processo de divisão da imagem em regiões (partes ou grupos) disjuntas de *pixels* que melhor representam os objetos da imagem, de acordo com algum critério de similaridade [37,38,39,40,41]. Após a formação das regiões, estas são interpretadas e

analisadas [42] por um sistema em outro nível, como, por exemplo, um sistema de reconhecimento de objetos [43], de edição de imagens [44], de compressão e recuperação de imagens [45], de planejamento dos passos de um robô [46], de detecção de pessoas [47], de faces [47], de gestos e de impressões digitais [48], de rastreamento de alvos (este artigo), de localização de objetos em imagens *SAR* [49], de localização de tumores [50], de vigilância [11], de reconhecimento de placas de trânsito [51] e de estimativa da posição de objetos oclusos em sistemas de acompanhamento de alvo [52]. Todos os passos posteriores à segmentação dependem da qualidade do resultado gerado. A segmentação de imagens, portanto, serve de apoio para a solução de problemas que estão em um nível superior, dentro do processamento de imagens. No caso do rastreamento de alvos pela rede neural *WiSARD*, a segmentação serve para binarizar a imagem de cada quadro do vídeo, tornando possível o treinamento da rede e a busca do alvo.

O critério de similaridade empregado considera uma ou mais características intrínsecas como nível de cinza ou textura e depende da aplicação [53]. A característica presente em uma imagem é uma propriedade pertencente a um *pixel* ou a uma região de *pixels*. A maioria dos algoritmos de segmentação utiliza informações locais (geralmente da cor), em uma região de vizinhança *V* do *pixel* em estudo. O critério de similaridade geralmente é uma regra que é aplicada apenas em *V*. Regiões devem ser separadas por possuírem aspectos diferentes.

O segmentador [37] deve considerar o contexto e obter conhecimentos prévios sobre a imagem para melhorar a desempenho da detecção de objetos. Para rastrear alvos de superfície, toda a região aérea deve ser descartada. Objetos aéreos de cor próxima da de um navio não devem ser rastreados.

Todo algoritmo de rastreamento de alvos deve atender dois requisitos básicos: rapidez e precisão. A evolução dos computadores facilita a construção de algoritmos de segmentação que atendem estes requisitos mesmo sendo mais complexos. Além destes, os algoritmos devem atender outros requisitos que dependem da aplicação. Por exemplo, o algoritmo de rastreamento de alvos de superfície não precisa ser tão rápido quanto o de acompanhamento de um avião que faz manobras no ar. O algoritmo de mapeamento de ruas, instalações e quartéis inimigos em uma imagem de satélite precisa localizar muitas linhas finas e retas enquanto que o de mapeamento de um navio deve ser preciso também nas curvas que acompanham a borda do navio. A imagem a ser processada geralmente possui *pixels* ruidosos devido a fatores ambientais ou fatores intrínsecos ao *hardware* dos equipamentos. A presença de ruído, de sombras, de textura, da transição pequena e gradual entre regiões adjacentes, de muitos objetos próximos e com cores similares, de variação na iluminação e da oclusão total ou parcial do objeto alvo são fatores que dificultam a segmentação. Antes de a segmentação ser feita, a imagem é pré-processada com filtros para diminuir a influência destes fatores.

Não existe um algoritmo de segmentação que atenda adequadamente todas as aplicações [41] de forma eficiente. Existem algoritmos que atendem uma gama maior de aplicações do que outros, porém, quanto mais especializado e consequentemente menos abrangente for o algoritmo, mais eficiente será para a aplicação. Até um algoritmo extremamente sofisticado pode apresentar um resultado pobre em uma determinada aplicação se não for adaptado.

A grande quantidade de algoritmos existentes torna desafiadora a tarefa de classificar os algoritmos em categorias. Alguns autores categorizam em: baseados em região, em detecção de borda, em limiarização, em segmentação por *clusters* e os híbridos [54]. Outros dividem apenas nos baseados em bordas e nos baseados em regiões [55,39,46,49]. Já [56] separa em três perspectivas: região, borda e região de transição. [57] enumera três categorias: baseados em detecção de borda, em *clustering* e em regiões. Existem aqueles [38] que separam em baseados em cor e baseados em textura. [55] divide em algoritmos baseados em limiarização, em reconhecimento de padrões e em modelos deformáveis. [45] classifica em baseados em bordas, em regiões ou em *clusters*. Independente da categoria do algoritmo, todos devem atender eficientemente o objetivo da aplicação.

Hoje existem na literatura muitos modelos de representação da cor. Não existe um modelo que seja o melhor para todas as situações e aplicações. Por exemplo, após o por do sol, os objetos da cena ficam nítidos em imagens infravermelhas. Porém, durante o dia, imagens RGB discriminam melhor os objetos do que imagens infravermelhas. Para desenvolver o rastreador baseado na rede neural sem peso *WiSARD* foi feito um estudo comparativo de diversos modelos para verificar qual é o mais adequado para a detecção de alvos de superfície em alto mar. Em todos os vídeos testados, os *pixels* inicialmente estão no modelo *RGB*. Antes de binarizar (passo 2 do rastreamento – seção 2.1) os *pixels* foram convertidos do modelo *RGB* para outro modelo. Com a ajuda de dois limiares utilizados em cada um dos canais do modelo transformado (no total são 6 limiares para todos os modelos exceto o de tons de cinza), o *pixel* é binarizado. Os modelos testados foram: tons de cinza, *HSI* (*hue*-H, saturação-S e intensidade-I) [36], *RGB* (vermelho-R, verde-G e azul-B), *HSV* (*hue*-H, saturação-S e valor-V)[36] e *YCbCr* (*Luma*-Y, *Blue Chroma*-Cb, *Red Chroma*-Cr) [58].

No modelo de tons de cinza, cada *pixel* da imagem assume valores dentro do intervalo [0..255]. Um *pixel* de valor 0 representa

a cor preta. O valor 255 representa a cor branca. Um valor intermediário representa a cor cinza em intensidades diferentes. *Pixels* representados pelo modelo *HSI* tem sua cor composta por três componentes: *hue*, saturação e intensidade. Estas três componentes podem ser calculadas a partir dos valores do modelo *RGB*. A componente *hue* é associada ao comprimento de onda de luz visível predominante. A saturação indica o grau de pureza relativa de uma cor. Toda cor é a mistura de um matiz com a componente branca. As cores puras possuem 0% de branco misturado com o matiz, sendo, portanto, completamente saturadas. Por exemplo, a cor rosa (vermelho com branco) é menos saturada do que a cor vermelha pura. Saturação e *hue* juntas são denominadas cromaticidade. O conjunto composto pela cromaticidade e pela intensidade caracteriza completamente uma cor. O modelo *HSI* separa a cromaticidade da intensidade da cor de forma semelhante à percepção de cores pelo ser humano, sendo, portanto, o melhor modelo para desenvolver algoritmos de segmentação de imagens baseados no funcionamento do sistema visual humano. O modelo *RGB* surgiu a partir da tentativa de modelar o sistema de cones (fotorreceptores do sistema visual humano). A cor de cada *pixel* é a mistura de três componentes: vermelho (R), verde (G) e azul(B). Geralmente cada componente assume valores inteiros de 0 até 255. O modelo *RGB* é usado em monitores coloridos e em câmeras de vídeo. O modelo *HSV* é semelhante ao modelo *HSI* exceto pela inclusão da componente *V* (valor) que substitui a componente *I* (intensidade). No modelo *YCbCr* as componentes *RGB* são transformadas para representar informações perceptualmente significativas. São utilizadas no processamento de imagens nas câmeras fotográficas e de vídeo digitais, e nas televisões *HDTV*.

Um estudo realizado por Woebbecke em 1995 [36] demonstrou que os modelos que possibilitam uma segmentação de melhor qualidade são o modelo *2g-r-b* (modelo *RGB* cuja componente verde possui maior peso) e o modelo denominado *hue* modificado, que explora a componente *hue*. De todas as componentes de cor dos diversos modelos, de acordo com [59] a que menos é influenciada pela variação de luminosidade, presença de sombra e de alguns tipos de brilho é a componente *hue*. Porém, dentre os modelos testados, o *YCbCr* foi o que melhor separou o alvo de superfície em alto-mar do cenário de fundo nos quadros dos vídeos de teste (seção 4.4.2). As componentes *Y*, *Cb* e *Cr* foram calculados de acordo com a equação 1.

$$\begin{aligned} Y &= 0{,}299.R + 0{,}587.G + 0{,}114.B \\ Cb &= 0{,}56433.(B - Y) \\ Cr &= 0{,}71327.(R - Y) \end{aligned} \qquad (1)$$

3.1 Principais métodos de segmentação

Nesta seção está descrito resumidamente o funcionamento dos métodos de segmentação mais conhecidos. Dependendo da aplicação, algumas destas técnicas são usadas em conjunto [46] para melhorar a acurácia, a robustez e a performance. As vantagens de um compensam os problemas do outro. A tendência atual é o desenvolvimento de métodos híbridos eficientes.

Limiarização [37,60,44,56,61] é uma das técnicas mais populares devido a simplicidade [62]. É baseada na hipótese de que os objetos podem ser distinguidos pelo nível de cinza ou cor dos *pixels*. As regiões são separadas com o auxílio dos valores de limiares. É denominada binível [57,62] quando apenas 1 limiar é utilizado. Neste caso os *pixels* são classificados em dois grupos (binarização): o objeto e o fundo da imagem [57]. Um grupo contém *pixels* com nível de cinza maior que o limiar e o outro contêm *pixels* com nível de cinza menor que o limiar. Já na segmentação multinível [62], a binarização utiliza dois ou mais valores de limiares. *Pixels* com nível de cinza pertencentes à alguns dos intervalos definidos pelos limiares são classificados em um grupo e os *pixels* restantes são classificados em outro grupo. Muitos autores utilizam o histograma da imagem para calcular os limiares [57]. Uma desvantagem da limiarização é que este método desconsidera a informação espacial dos *pixels*. Os algoritmos automáticos de cálculo de limiares mais conhecidos são: limiarização convencional [44], busca otimizada por Algoritmos Genéticos [60], limiarização por *P-tile* [61], erro mínimo [63], limiarização por amostragem [64], método de Otsu [65], método de Niblack [66], método de Rosin [67] e método da entropia máxima [68].

Os algoritmos baseados em detecção de borda [38,39,40,61] baseiam-se no fato das bordas guardarem a informação do local onde objetos diferentes se encontram. As bordas são locais cujos *pixels* são descontínuos em relação a algumas características, como, por exemplo, a cor. Os objetos de interesse ficam separados dos demais pelas suas bordas [55]. A detecção de borda utiliza-se de informações locais. Dois operadores são amplamente utilizados para a detecção de bordas: o gradiente e o Laplace [61]. O gradiente e o Laplace podem ser calculados percorrendo-se uma máscara sobre os *pixels* da imagem. Máscaras como Roberts, Sobel e Prewit são usados para calcular o gradiente. As máscaras que servem para o cálculo do Laplace e do gradiente estão apresentadas na figura 4. Alguns algoritmos aplicam estes operadores juntamente com o uso de um limiar para a detecção de bordas [39]. O método descrito por [69] detecta as bordas somente com autovetores. Já a detecção de borda proposta por John F. Canny [70] é executado em vários estágios e em apenas um destes estágios utiliza-se uma máscara (operador Sobel).

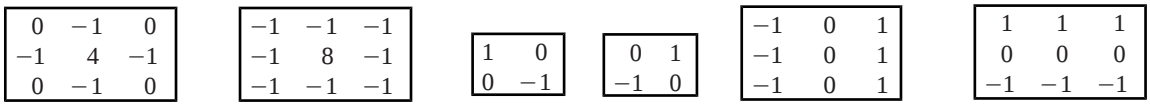


Figura 4: Máscaras para o cálculo do Laplace e do gradiente: (a) Laplace 4 conectada; (b) Laplace 8 conectada; (c) Gradiente a 135^{o} por Roberts; (d) Gradiente a 45^{o} por Roberts; (e) Gradiente em x por Prewitts; (f) Gradiente em y por Prewitts;

A segmentação por crescimento de regiões utiliza um critério de similaridade de um ou mais aspectos (*features*) entre *pixels* vizinhos [55,44,56,45,61]. Por hipótese os *pixels* dos objetos de interesse são semelhantes em relação aos aspectos escolhidos. Os *pixels* são agrupados recursivamente a partir de um ou mais *pixels* denominados sementes de onde se inicia o processo de crescimento das regiões. *Pixels* vizinhos que atendem ao critério são agregados à região iniciada por uma semente. O crescimento continua até que o critério de similaridade não seja atingido ou até que a região de crescimento de outra semente seja alcançada. Pelo critério de seleção das sementes, o método de crescimento de regiões pode ser dividido em *Seeded Region Growing* e *Unseeded Region Growing*. A principal diferença é que naquele a escolha das sementes iniciais é manual enquanto que neste a escolha é automática. O número de sementes pode se manter constante durante a segmentação ou aumentar quando houver a necessidade de criação de outra região pelo não atendimento do critério de similaridade de algum *pixel*.

O método *Watershed Transformation* foi introduzido por [71]. É considerado por alguns autores como um método híbrido de crescimento de regiões com detecção de bordas. A segmentação é baseada na morfologia matemática [41,50] e geralmente é executada sobre uma imagem de gradientes [61,71], mas pode ser executada sobre imagens de tons de cinza [44,45,50,72,61] ou sobre uma imagem transformada [73]. A idéia inicial surgiu a partir de uma observação: locais onde o gradiente é elevado estão as bordas dos objetos e onde são pequenos estão seus interiores. No caso de uma imagem de níveis de cinza, observa-se o fato de cada objeto possuir um nível diferente dos demais. Fazendo uma analogia com imagens topográficas [44,45,50,73], o gradiente está relacionado à altura do relevo. Superfícies planas que possuem pelo menos um ponto de altura mínima (*LMI-local minima intensity*) são denominadas vales (*catchment basins*). São rodeadas por montanhas cujos cumes são denominados *dam*. Os *dam* indicam locais de gradiente localmente máximos. A segmentação inicia nos *LMI*. Quando uma chuva começa, os *LMI* são imediatamente inundados. Em cada vale, a água atinge primeiramente o ponto de menor altura. Posteriormente a água invade o outro lado da montanha que rodeia o vale. Para evitar a invasão, muros denominados *watershed lines* são criados nestes locais delimitando as regiões da imagem. A segmentação continua até que a imagem contenha apenas *watershed lines* e água. Os algoritmos mais conhecidos são o de Meyer [72], de Vicent-Soile [44], o baseado em filas [50] e o de Belhomme[74].

O método *Split and Merging* é simples, efetivo e não supervisionado. A idéia é dividir regiões em sub-regiões caso seus *pixels* sejam heterogêneos em uma primeira fase (*top-down*) e em uma segunda fase (*botton-up*) caso duas regiões adjacentes forem homogêneas, elas são unidas para formar uma única região. Os agrupamentos se repetem sucessivamente até que todas as regiões sejam suficientemente heterogêneas. A ordem das divisões e dos agrupamentos varia de acordo com o algoritmo. Na forma mais simples, a imagem é considerada inicialmente como tendo apenas uma região [45,61]. Diversos autores [56] relatam técnicas para superar as dificuldades e problemas encontrados.

Dentre os métodos de segmentação comparados para o projeto do rastreador proposto neste artigo, o mais eficiente (rápido e preciso) foi um método hibrido inovador que realiza a binarização por limiarização, onde os limiares são calculados com o auxilio de um modelo Gaussiano e os *pixels* são amostrados por detecção de borda em duas direções (seção 4.2). Uma abordagem mais eficiente deste método inovador está descrito na seção 4.4.2.

4 Projeto do rastreador

Nesta seção são abordadas as simulações realizadas com o rastreador baseado na rede neural sem peso *WiSARD* proposto. As simulações objetivam modificar a configuração do rastreador para melhorar sua performance.

4.1 Características dos vídeos de teste

Todos os vídeos são artificiais (figura 5), construídos com imagens reais de navios, mar e céu. Todos os quadros têm resolução de 640x480 *pixels*. O cenário tático contém apenas um alvo de superfície com velocidade e aceleração muito maiores que um navio real para dificultar o rastreador. Os alvos de superfície reais mudam pouco de posição (mudam muito menos do que o valor do seu comprimento) em quadros consecutivos quando uma câmera estabilizada é colocada para capturar o cenário tático. Os módulos dos vetores de velocidade e de aceleração dos alvos simulados estão detalhados na tabela 1. Os nomes dos vídeos começam com a letra *V*. A classe do navio é indicada pelas letras que seguem a letra *V*. Nomes diferenciados apenas por

números no final contêm o mesmo cenário e navio alvo, porém este realiza manobras diferentes em cada um.

Tabela 1: características dos vídeos de teste. O sinal + significa muito. O módulo do vetor velocidade é muito grande quando a diferença da posição do navio em quadros consecutivos é aproximadamente o dobro do seu comprimento. O módulo é grande quando esta diferença é aproximadamente igual ao comprimento do navio. O módulo é médio quando esta diferença é aproximadamente a metade do comprimento do navio. O módulo é pequeno quando esta diferença é menor que a metade do comprimento do navio.
O módulo do vetor aceleração é muito grande quando em determinados pontos da trajetória o navio muda o valor do ângulo do vetor velocidade em aproximadamente 180^{o} em quadros consecutivos. O módulo do vetor aceleração é grande quando o navio demora 2 a 3 quadros para mudar o módulo do vetor velocidade do valor máximo para zero. O módulo do vetor aceleração é médio quando o navio demora em torno de 6 quadros para mudar o módulo do vetor velocidade do valor máximo para zero. O módulo do vetor aceleração é pequeno quando o navio demora em torno de 10 quadros para mudar o módulo do vetor velocidade do valor máximo para zero.

Nome	*Classe do navio alvo*	*Número total de quadros*	*Área (pixels) da seleção inicial*	*Porte do alvo*	*Contraste entre alvo e mar*	*Contraste entre alvo e céu*	*Módulo do vetor velocidade*	*Módulo do vetor aceleração*	*Variação de rumo*
VF	***F***	55	40x17	pequeno	médio	pequeno	médio	pequeno	média
VB1	***B***	79	68x24	grande	pequeno	médio	médio	médio	grande
VB2	***B***	66	68x19	grande	pequeno	médio	grande	grande	grande
VZ	***Z***	87	42x38	grande+	médio	médio	pequeno	pequeno	pequena
VC	***C***	60	36x25	médio	médio	pequeno	médio	médio	grande
VS	***S***	50	61x21	médio	grande	grande	grande+	grande+	grande+
VP1	***P***	79	40x21	médio	grande	grande	médio	grande	grande
VP2	***P***	79	47x21	médio	grande	grande	grande	grande	grande
VUW1	***UW***	75	34x29	médio	pequeno	pequeno	médio	pequeno	média
VUW2	***UW***	28	55x26	médio	pequeno	pequeno	médio	médio	média
VAE	***AE***	78	80x27	grande	grande	grande	pequeno	pequeno	pequena

Figura 5: Vídeos de teste. Os vídeos VF, VB1 (e VB2), VZ, VC, VS, VP1 (e VP2), VUW1 (e VUW2) e VAE estão dispostos da esquerda para a direita, de cima para baixo.

4.2 Segmentação utilizada pelo rastreador proposto

A segmentação (binarização) em todas as simulações até a configuração de discriminadores *GP6* (seção 4.4.2) foram feitas com o modelo de cor *RGB* utilizando um método de segmentação híbrido inovador que envolve limiarização e detecção de

borda por ter sido o método mais eficiente entre os testados. Um aperfeiçoamento deste método está descrito na seção 4.4.2. Para a comparação, um conjunto de 40 imagens de teste contendo navios no mar foi segmentado manualmente. A segmentação manual (segmentação de referência) foi comparada com a segmentação resultante de cada método. As três funções utilizadas para comparar foram: precisão ([*NVP*] / [*NVP* + *NFP*]), quantidade relativa de *pixels* erroneamente segmentados (*NES* = [*NFN* + *NFP*]/[*N*]) e acurácia ([*NVP* + *NVN*]/[*N*]). As três funções [75] são supervisionadas, pois comparam o resultado da segmentação dos algoritmos com a segmentação manual de referência e quantitativas, pois expressam a qualidade em valores numéricos. *NVP*, *NVN*, *NFP*, *NFN* e *N* são respectivamente: número de *pixels* do alvo corretamente segmentados, número de *pixels* do fundo corretamente segmentados, número de *pixels* do alvo erroneamente segmentados, número de *pixels* do fundo erroneamente segmentados e número total de *pixels* da imagem. *N* é o mesmo para todas as imagens. Um método de segmentação é de melhor qualidade quando consegue obter o maior valor de precisão e acurácia e o menor valor de NES. Nenhum método de segmentação consegue superar os demais para todas as imagens de teste. Porém, foi escolhido o método de segmentação que obteve a melhor performance em pelo menos duas das três medidas quantitativas no maior número de imagens de teste. Os métodos testados foram: limiarização, detecção de borda, limiarização com detecção de borda, crescimento de regiões, *Watershed Transformation* e *Split and Merging*.

1- O centro da janela de seleção inicial *cs* realizada pelo operador sempre pertence ao alvo de superfície a ser rastreado. A detecção de quatro pontos que pertencem às bordas, *p1*, *p2*, *p3* e *p4* são calculados deslizando o filtro detector de borda Prewitts na direção transversal e longitudinal ao navio em quatro sentidos distintos partindo de *cs*: para cima (*p1*), para baixo (*p2*), para a esquerda (*p3*) e para a direita (*p4*) (figura 6). Estes quatro pontos definem duas faixas de *pixels* que pertencem ao alvo: uma faixa transversal limitada por *p1* e *p2* e uma faixa longitudinal limitada pelos pontos *p3* e *p4* (figura 6-c). Os *pixels* contidos nas duas faixas, *pixels pf*, pertencem ao alvo e são uma boa amostra de *pixels* para o cálculo dos limiares.

A vantagem de representar o alvo por amostragem de seus *pixels* em duas faixas é a rapidez da amostragem e consequentemente do cálculo dos parâmetros do segmentador, essencial para o rastreamento de alvos. A utilização de um método padrão de segmentação para separar todos os *pixels* pertencentes ao alvo para depois calcular os parâmetros do segmentador é computacionalmente mais custoso. Os *pixels* contidos nas duas faixas são suficientes para representar o alvo, pois geralmente os navios apresentam cores sem muita variação de tonalidade.

2- Calcular a mediana e o desvio padrão dos *pixels pf* para os três canais *R*, *G* e *B*. A média não foi utilizada, pois esta é mais influenciada por *pixels* ruidosos.

3- Amostrar os *pixels pw* pertencentes ao mar e calcular a mediana para os três canais *R*, *G* e *B*.

4- Calcular seis limiares, dois para cada canal do modelo *RGB*. Cada par de limiares delimita um dos 3 canais *RGB* de forma a incluir os *pixels* pertencentes ao alvo e excluir os *pixels* pertencentes ao cenário de fundo:

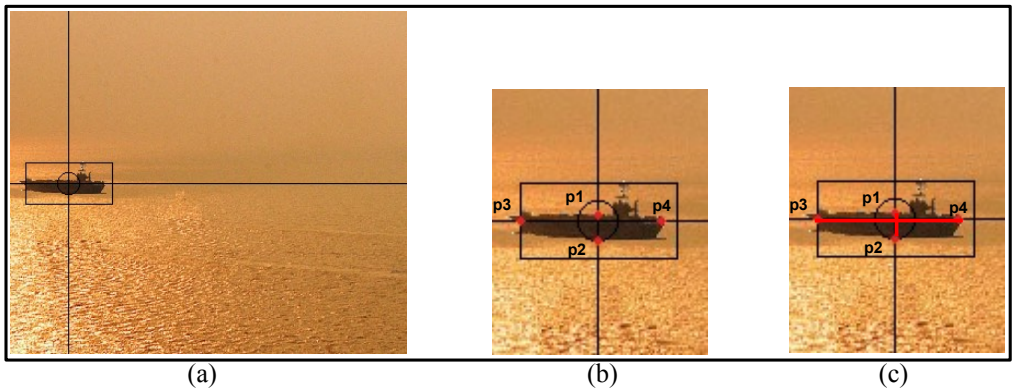
$$\begin{aligned} L1 &= medianapfR + x.desviopfR \\ L2 &= medianapfR - x.desviopfR \\ L3 &= medianapfG + y.desviopfG \\ L4 &= medianapfG - y.desviopfG \\ L5 &= medianapfB + z.desviopfB \\ L6 &= medianapfB - z.desviopfB \end{aligned} \quad (2)$$

Os valores *x*, *y* e *z* são escalares que permitem *separar os pixels pf dos pixels pw*. Nem sempre é possível separar eficientemente por este método. Por este motivo, uma abordagem mais eficiente é descrita na seção 4.4.2.

5- Binarizar a imagem de cada quadro posterior ao primeiro utilizando-se estes limiares.

4.3 Janela de seleção e de busca simulados

A única ação realizada pelo operador é selecionar uma área do primeiro quadro que contenha o alvo, definindo a janela de seleção. Os *pixels* da janela de seleção são binarizados e utilizados para treinar todos os discriminadores da rede neural sem peso *WiSARD*. A seleção inicial ideal é aquela que contem tanto o navio alvo quanto o cenário de fundo para que os discriminadores sejam treinados com os dois padrões. O rastreamento do alvo depende da seleção inicial feita pelo operador. Para que a seleção inicial não influencie nos resultados das simulações, as seleções iniciais dos vídeos foram mantidas constantes em todas as simulações. O tamanho da janela de busca depende da configuração dos discriminadores. Ela é a união das áreas de cada quadro cobertas por cada discriminador. Em cada simulação realizada (seção 4.4) foi utilizada uma configuração diferente.



(a) (b) (c)

Figura 6: (a) janela de seleção inicial; (b) pontos *p1*, *p2*, *p3* e *p4*; (c) duas faixas de *pixels* limitadas por *p1*, *p2*, *p3* e *p4*.

4.4 Simulações realizadas

Todos os discriminadores da rede neural possuem nós *RAM* de 9 *bits* nas simulações apresentadas nas seções 4.4.1 e 4.4.2. Na seção 4.4.3 foi realizada uma busca pelo tamanho de nós *RAM* mais adequado para o rastreamento. A posição do centro geométrico da área coberta por cada discriminador fica deslocado do centro geométrico das áreas vizinhas cobertas por outros discriminadores de *xp pixels* em relação ao eixo *x* da imagem e *yp pixels* em relação ao eixo *y* da imagem. Os centros geométricos das áreas cobertas por todos os discriminadores formam uma grade de pontos *GP*.

4.4.1 Configuração dos discriminadores da rede *WiSARD*

A simulação *S1* objetiva a escolha da quantidade ideal de discriminadores da rede neural *WiSARD*. Existe uma relação de custo e benefício. Quanto maior for a quantidade de discriminadores maior será a janela de busca, menor será a probabilidade de perder o alvo, mas maior será o tempo médio *PR* de execução do algoritmo de rastreamento em um quadro. O erro de rastreamento ocorre quando o rastreador aponta para uma posição onde o alvo não está presente. A Tabela 2 apresenta a relação de tempo médio *PR* x quadro que ocorreu o primeiro erro de rastreamento.

Tabela 2: simulação *S1*: relação de tempo médio de rastreamento *PR* x quadro que ocorreu o primeiro erro de rastreamento. *GP1* possui 12 x 12 discriminadores espaçados de 2 *pixels xp* e *yp* em relação aos seus vizinhos. *GP2* possui 20 x 20 discriminadores espaçados de 2 *pixels xp* e *yp* em relação aos seus vizinhos. *GP3* possui 30 x 30 discriminadores espaçados de 2 *pixels xp* e *yp* em relação aos seus vizinhos. *GP4* possui 20 x 20 discriminadores espaçados de 5 *pixels xp* e *yp* em relação aos seus vizinhos. *GP5* possui 20 x 20 discriminadores espaçados de 5 *pixels xp* e *yp* em relação aos seus vizinhos e o rastreador possui um preditor simples de posição. *N* presente na última coluna indica que o rastreador não cometeu erro de rastreamento até o final do vídeo.

Nome do vídeo	***PR(ms) com GP1***	***PR(ms) com GP2***	***PR(ms) com GP3***	***PR(ms) com GP4***	***PR(ms) com GP5***	***Quadro do 1º erro com GP1***	***Quadro do 1º erro com GP2***	***Quadro do 1º erro com GP3***	***Quadro do 1º erro com GP4***	***Quadro do 1º erro com GP5***
VF	50	101	139	83	82	35º	35º	32º	3º	3º
VB1	102	224	351	233	242	42º	64º	70º	N	N
VB2	105	203	318	214	220	63º	70º	N	N	N
VZ	121	269	427	253	267	N	N	N	N	N
VC	51	91	161	109	105	53º	N	N	29º	19º
VS	85	165	240	176	190	3º	3º	3º	N	58º
VP1	123	254	391	250	257	34º	71º	73º	77º	N
VP2	155	333	493	352	355	32º	70º	73º	78º	N
VUW1	65	129	247	130	130	67º	N	N	71º	N
VUW2	82	181	252	159	184	17º	24º	26º	13º	23º
VAE	104	239	412	239	242	N	N	N	N	N

OBSERVAÇÕES:

1- Tempo médio *PR*: quanto maior a quantidade de discriminadores da rede *WiSARD*, maior é o tempo médio *PR*.

2- Quanto maior é a quantidade de discriminadores menor é a probabilidade de erro. Porém, o rastreador errou o alvo 3 quadros antes com *GP3* quando comparado a *GP1* e *GP2* devido a erros de binarização no vídeo *VF*. A resposta da rede *WiSARD* é maior em uma posição que só contém água. Em *VS* o alvo não é rastreado, perdendo-o no 3º quadro com *GP1*, *GP2* e *GP3*. O alvo faz manobras não reais (velocidade muito alta e mudança instantânea de rumo), saindo da janela de busca. Porém, com *GP4* não há erros até o final do vídeo, pois houve aumento da área coberta pelos discriminadores.

3- Com a binarização mais exata o rastreador comete menos erros. Em *VF*, *VC*, *VUW1* e *VUW2*, *GP4* proporciona desempenho inferior se comparado com *GP1*, *GP2* e *GP3* com relação ao quadro que ocorre o primeiro erro apesar da janela de busca ser maior devido a erros de binarização dos *pixels* da janela de seleção. Com *GP5* o desempenho é inferior se comparado à *GP4* em *VF* e *VC* pelo mesmo motivo. O preditor posiciona a janela de busca em uma região duvidosa alguns quadros antes.

4- A introdução de um preditor simples que calcula a aceleração e a velocidade do alvo para prever a posição no próximo quadro melhora a performance do rastreador na maioria dos vídeos. Porém, com *GP5* o rastreador comete o primeiro erro antes se comparado com *GP4* em *VS* devido ao movimento imprevisível do alvo que engana o preditor. Este posiciona a janela de busca longe do alvo (figura 7). O preditor deve ser desligado quando não previr a posição do alvo corretamente.

5- Considerando *PR* e a performance com relação ao quadro que ocorre o primeiro erro, a configuração *GP5* foi mais eficiente mostrando a importância do preditor.

4.4.2 Evolução do método hibrido de binarização proposto

Utilizando a configuração *GP5*, o rastreador perde o alvo antes do final dos vídeos *VF*, *VC*, *VS* e *VUW2*. Para melhorar a performance, duas configurações *GP6* e *GP7* que possuem duas camadas de discriminadores foram testadas (tabela 3). Por hipótese, aumentando a densidade de discriminadores perto da posição indicada pelo preditor, o rastreador cometerá menos erros. A binarização utilizada em *GP6* (seção 4.2) é a mesma que a utilizada nas configurações apresentadas na seção 4.4.1. Já a utilizada em *GP7* é uma evolução deste método. Diversos métodos de segmentação e modelos de cor foram testados. A combinação do modelo de cor *YCbCr* com um método híbrido de segmentação foi a que obteve os melhores resultados, sendo utilizado em todas as configurações a partir de *GP7*. Algoritmo proposto modificado:

1- Calcular os *pixels pf* (passo 1, seção 4.2), transformar do modelo *RGB* para o modelo *YCbCr* (seção 3). Posteriormente, calcular a mediana e o desvio padrão dos *pixels pf* para os três canais *Y*, *Cb* e *Cr*.

2- Calcular seis limiares pela equação 3 (equivalente ao passo 4, seção 4.2). Em todos os vídeos, os parâmetros x=3, y=1.5 e z=3 são suficientes para eliminar os erros de rastreamento casados por uma binarização ineficiente (tabela 3). Nos quadros posteriores ao primeiro, os *pixels* da janela de busca são transformados para o modelo *YCbCr* e binarizados com estes limiares.

$$
\begin{aligned}
L1 &= mediana pf Cr + x.desvio pf Cr \\
L2 &= mediana pf Cr - x.desvio pf Cr \\
L3 &= mediana pf Y + y.desvio pf Y \\
L4 &= mediana pf Y - y.desvio pf Y \\
L5 &= mediana pf Cb + z.desvio pf Cb \\
L6 &= mediana pf Cb - z.desvio pf Cb
\end{aligned}
$$

A introdução de uma segunda camada de discriminadores, *GPD*, mais densa e menor que *GP*, melhorou o rastreamento nos vídeos *VF*, *VC* e *VS*, porém, no vídeo *VUW2* com a configuração *GP6*, a resposta da rede neural *WiSARD* é maior em uma posição que só contém água do mar 9 quadros antes do que com *GP5* por erro de binarização. Os vídeos *VUW1*, *VUW2*, *VB1* e *VB2* são muito desafiadores, pois a cor do alvo se confunde com a cor do mar, produzindo mais erros de binarização. A configuração *GP7* por utilizar um método mais eficiente e inovador de binarização consegue rastrear o alvo até o fim de todos os vídeos mostrando a eficiência deste método para o rastreamento de alvos de superfície no mar e a influência da binarização correta no rastreamento.

Tabela 3: simulação *S2*: relação de tempo médio de rastreamento *PR* x quadro que ocorreu o primeiro erro de rastreamento. *GP5* é a mesma configuração da simula*ção* S1 (tabela 2). *GP6* e *GP7* possuem duas camadas de discriminadores: a primeira, *GP*, possui 20 x 20 discriminadores espaçados de 5 *pixels xp* e *yp* em relação aos seus vizinhos; a segunda, *GPD*, mais densa que *GP* possui 10 x 10 discriminadores espaçados de 2 *pixels xp* e *yp* em relação aos seus vizinhos. Ambas as camadas são centralizadas na posição do alvo prevista pelo preditor. *GP5* e *GP6* utilizam a segmentação apresentada na seção 4.2 e *GP7* utiliza a segmentação apresentada nesta seção. *N* presente na última coluna indica que o rastreador não cometeu erro de rastreamento até o final do vídeo.

Nome do vídeo	***PR(ms) com GP5***	***PR(ms) com GP6***	***PR(ms) com GP7***	***Quadro do 1^{o}erro com GP5***	***Quadro do 1^{o}erro com GP6***	***Quadro do 1^{o}erro com GP7***
VF	82	95	93	3^{o}	21^{o}	N
VB1	242	290	287	N	N	N
VB2	220	275	259	N	N	N
VZ	267	318	326	N	N	N
VC	105	107	107	19^{o}	N	N
VS	190	227	236	58^{o}	N	N
VP1	257	305	309	N	N	N
VP2	305	358	364	N	N	N
VUW1	130	146	152	N	N	N
VUW2	184	209	218	23^{o}	14^{o}	N
VAE	242	297	300	N	N	N

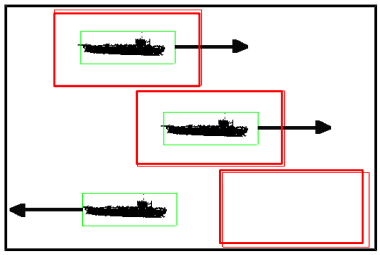


Figura 7: Exemplo de movimento irregular. A mudança de velocidade do alvo engana o preditor, colocando a janela de busca (em vermelho) para um local onde o alvo não se encontra. Quadro 1 está acima, quadro 2, no centro e o quadro 3, abaixo.

4.4.3 Tamanho dos nós *RAM* ideal

Buscando diminuir *PR*, foi utilizando um rastreador com a configuração *GP7* para verificar qual o tamanho dos nós *RAM* que permite um rastreamento com menor *PR*, mantendo a mesma eficiência de não perder o alvo durante o rastreamento.

O número de situações realizadas foi muito grande de forma que é impossível colocar em uma tabela. Todos os vídeos foram testados com tamanho de nó *RAM* variando de 1 *bit* de endereçamento ao maior número de *bits* permitido pela memória do computador. A figura 8 mostra a relação de *PR* com o tamanho dos nós. A forma do gráfico é a mesma em todos os testes, mudando somente o tamanho de nó em que *PR* começa a aumentar muito. Este tamanho variou de 16 a 22 *bits* de endereçamento. *PR* aumenta pois o limite da memória *RAM* do computador é excedido.

Buscando estudar como o tamanho dos nós *RAM* da rede neural *WiSARD* pode compensar os erros de binarização de cada quadro e melhorar o desempenho do rastreador, a qualidade da binarização foi propositadamente piorada em vários graus modificando os valores dos limiares *L1* a *L6*. Todos os tamanhos de nó *RAM* suportados foram testados.

Considerando somente *PR*, nós entre 2 e 14 *bits* de endereçamento possuem desempenho semelhante, porém, nós com 3 *bits* de endereçamento são os que compensam melhor erros de binarização, apresentando melhor desempenho em relação ao quadro que ocorre o primeiro erro de rastreamento em cerca de 70% dos casos. Conclui-se que o tamanho dos nós influencia a

robustez do rastreador contra erros na binarização.

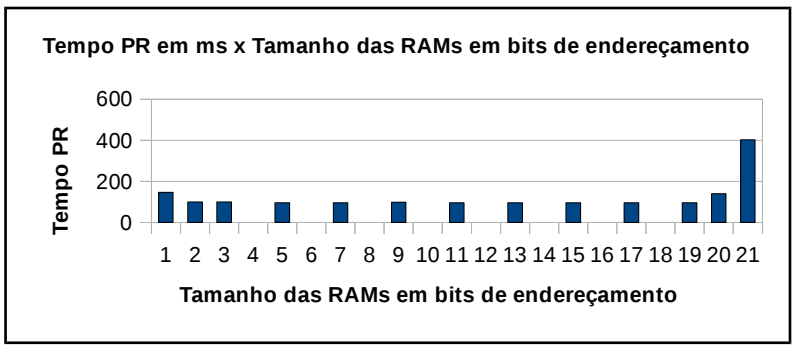


Figura 8: Relação de tempo médio *PR* x tamanho dos nós *RAM* para o vídeo *VF*. Alguns tamanhos de *RAM* não foram testados para calcular *PR*, pois logo nas primeiras simulações foi notado que os valores *PR* são sempre bem próximos aos obtidos com os tamanhos de *RAM* vizinhos (inteiro imediatamente superior e inferior). A não simulação resultou em economia de tempo. Porém, estes tamanhos de *RAM* foram utilizados para encontrar o tamanho que melhor compensa erros de binarização.

4.4.4 Quantidade de discriminadores

Os vídeos *VF*, *VC* e *VUW1* foram utilizados para verificar se existe uma relação entre densidade/quantidade de discriminadores e robustez contra erros de binarização. Nestes vídeos a qualidade da binarização foi propositadamente piorada em vários graus modificando os valores *L1* a *L6*. Quatro configurações de discriminadores foram testadas. Os vídeos *VF* e *VUW1* foram escolhidos por produzir mais binarizações erradas com uma menor variação dos limiares. *VC* foi escolhido por ter um navio cruzeiro no horizonte semelhante ao alvo. Devido a conclusão obtida na seção anterior, a rede neural *WiSARD* dos rastreadores testados possuem nós *RAM* com 3 *bits* de endereçamento.

Observando a tabela 4, conclui-se que existe uma relação entre o posicionamento dos discriminadores e a robustez contra erros de binarização. Comparando *GP8* com *GP9* e *GP10* com *GP11*, percebe-se que o rastreador foi mais eficiente utilizando discriminadores com espaçamento *xp* e *yp* iguais a 1 *pixel* (superou 12 vezes) do que com 2 *pixels* (superou apenas 1 vez), apesar da primeira utilizar menos discriminadores (25%). Um rastreador supera o outro quando acompanha o alvo por mais quadros. *GP11* foi a mais eficiente, superando as outras em 8 dos 9 testes. Conclui-se que existe uma relação entre a forma de distribuição dos discriminadores e a robustez do rastreador contra erros de binarização.

4.4.5 Rastreador com duas redes neurais *WiSARD* em paralelo

A rede *WiSARD* reconhece melhor um padrão pré treinado (a resposta é maior) quando os *pixels PO* que pertencem ao objeto são binarizados com um valor e os *pixels PNO* que não pertencem ao objeto são binarizados com outro. Caso um subconjunto de *PNO* seja binarizado erroneamente, a rede irá responder com uma soma menor. Na maioria dos métodos de segmentação, a probabilidade de um *pixel PNO* ser binarizado erroneamente é maior do a de um *pixel PO*, pois os métodos tem como objetivo principal incluir os *pixels PO* em uma região *RG* quando atendem requisitos *RQ*. O foco da segmentação é a inclusão dos *pixels* de interesse (*PO*) em *RG*. Excluir os *pixels PNO* de *RG* e incluir em outra é apenas uma consequência do não atendimento de *RQ*. A exclusão é um dos objetivos, mas não é o principal. Intrinsecamente, excluir um *pixel PO* de *RG* é mais grave do que incluir um *pixel PNO* em *RG*. Isto é observado pela forma que os métodos definem *RQ*.

O rastreador proposto neste artigo inclui os *pixels PO* em *RG* caso sua cor pertença a um intervalo definido pelos limiares *L1* a *L6*. A exclusão dos *pixels PNO* do intervalo definido pelos limiares não é a prioridade, é apenas uma consequência. A exclusão ocorre apenas se os *pixels PNO* são suficientemente diferentes aos *pixels PO*.

Qual é a influência de cada erro de binarização na resposta da rede neural? Caso o mapeamento de entrada dos nós *RAM* seja criado aleatoriamente, o erro de binarização de um subconjunto com *X pixels PO* acarreta a mesma diminuição no valor da resposta da rede neural *WiSARD* que o erro de binarização de um subconjunto com *X pixels PNO*. Mas como o objetivo principal da maioria dos métodos de segmentação é a binarização correta dos *pixels PO* e não a binarização correta dos *pixels PNO*, então existe uma maior probabilidade da ocorrência de erros de binarização dos *pixels PNO* do que dos *pixels PO*.

Tabela 4: relação entre o aumento percentual do número relativo de *pixels* binarizados erroneamente dentro da janela de seleção x quadro que ocorreu o primeiro erro de rastreamento. *GP8* é composta por *GP* e *GPD1*. *GP9* é composta por *GP* e *GPD2*. *GP10* é composta por *GP*, *GPD3*, *GPD4*, *GPD5* e *GPD6*. *GP11* é composta por *GP*, *GPD7*, *GPD8*, *GPD9* e *GPD10* . *GP* possui 20 x 20 discriminadores espaçados de 5 *pixels xp* e *yp* em relação aos seus vizinhos. *GPD1*, *GPD3*, *GPD4*, *GPD5* e *GPD6* possuem 10 x 10 discriminadores espaçados de 2 *pixels xp* e *yp* em relação aos seus vizinhos. *GPD2*, *GPD7*, *GPD8*, *GPD9* e *GPD10* possuem 5 x 5 discriminadores espaçados de 1 *pixel xp* e *yp* em relação aos seus vizinhos. As camadas *GP*, *GPD1*, *GPD2*, *GPD3* e *GPD7* são centralizadas na posição do alvo prevista pelo preditor. *GPD4* e *GPD8* são centralizadas na posição do discriminador de maior resposta de *GP*; *GPD5* e *GPD9* são centralizadas na posição do discriminador de segunda maior resposta de *GP*; e *GPD6* e *GPD10* são centralizadas na posição do discriminador de terceira maior resposta de *GP*. *N* presente na última coluna indica que o rastreador não cometeu erro de rastreamento até o final do vídeo.

Nome do vídeo	***Aumento percentual relativo de pixels binarizados erroneamente***	***Quadro do 1^{o}erro com GP8***	***Quadro do 1^{o}erro com GP9***	***Quadro do 1^{o}erro com GP10***	***Quadro do 1^{o}erro com GP11***
VF	34%	46^{o}	N	21^{o}	N
VF	43%	33^{o}	52^{o}	31^{o}	34^{o}
VF	65%	15^{o}	15^{o}	15^{o}	18^{o}
VC	91%	N	N	N	N
VC	93%	N	N	N	N
VC	97%	19^{o}	53^{o}	19^{o}	53^{o}
VUW1	42%	72^{o}	73^{o}	31^{o}	73^{o}
VUW1	69%	31^{o}	71^{o}	13^{o}	73^{o}
VUW1	122%	14^{o}	20^{o}	15^{o}	20^{o}

Projetar uma rede onde os *pixels PO* tenham mais peso na resposta acarretará mais acertos. A rede *WiSARD*, por definição, não possui pesos na soma final da rede (resposta da rede). Para contornar este problema, duas redes, dispostas em paralelo, cada uma responsável por um conjunto de *pixels* disjuntos e com nós *RAM* de tamanhos diferentes podem ser utilizadas. Uma rede com nós *RAM* de 3 *bits* de endereçamento gera uma resposta de valor maior do que uma rede com nós *RAM* de 15 *bits* de endereçamento por possuir uma maior quantidade de nós. A rede com mais nós (3 *bits* de endereçamento) será responsável pela parte central da janela de seleção (onde provavelmente os *pixels PO* do alvo estão localizados) e a rede com menos nós (15 *bits* de endereçamento) será responsável pela parte periférica da janela de seleção (onde provavelmente os *pixels PNO* do fundo estão) (figura 9). O paralelismo se dá no nível dos discriminadores. Na posição (x,y), centro do retângulo verde da figura 9, existe um discriminador *C* da rede paralela cobrindo todos os *pixels* no interior deste retângulo. *C* é a combinação (união) do discriminador *A* da rede *WiSARD* que possui mais nós *RAM* e está centralizado em (x,y), com o discriminador *B* da rede *WiSARD* que possui menos nós *RAM,* também centralizado em (x,y). *A* cobre os *pixels* no interior do retângulo azul, *B* cobre os *pixels* no interior dos retângulos abóboras e *C* cobre os *pixels* no interior do retângulo verde (figura 9). O discriminador *C* da rede em paralelo retorna como resposta a soma das respostas dos discriminadores *A e B*. Assim, todos os discriminadores da rede *WiSARD* em paralelo são formados pela combinação de dois discriminadores centralizados na mesma posição, um de cada rede. Esta forma inovadora de colocar duas redes neurais *WiSARD* em paralelo para rastrear o alvo e melhorar o seu desempenho quando erros de binarização ocorrem é a maior contribuição deste artigo.

Variando a porcentagem relativa de *pixels* erroneamente binarizados, 129 simulações foram realizadas para verificar a porcentagem *P* de *pixels* da janela de seleção coberta pela rede neural com nós *RAM* de 3 *bits* de endereçamento e pela rede neural com nós *RAM* de 15 *bits* de endereçamento (1-*P*) para aumentar a robustez contra erros de binarização. Para estes testes foi utilizada a configuração de discriminadores *GP11* por ser a mais eficiente contra erros de binarização (seção 4.4.4).

As simulações mostraram que existe uma porcentagem *P* (ou intervalo de porcentagens) de *pixels* cobertos pela rede neural com nós *RAM* de 3 *bits* de endereçamento que melhora a performance quando ocorrem erros de binarização. Quando o valor ótimo de *P* é utilizado, duas redes neurais *WiSARD* em paralelo é sempre mais eficiente do que uma rede possuindo nós *RAM*

com 3 *bits* de endereçamento. *P* não é igual em todos os vídeos. Em *VF*, *P* é de 40% a 45%. Em *VC*, *P* é de 80% a 85%. Em *VUW1*, *P* é de 20%. O valor ótimo de *P* depende de fatores como o tamanho do alvo, da janela de seleção e do contraste do alvo em relação ao fundo. O cálculo *online* de *P* pode ser feito quando o grau de confiança *C* ficar abaixo de um limiar ou quando o preditor comete um grave erro.

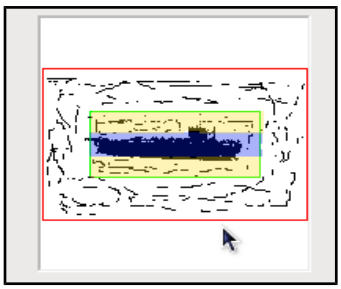


Figura 9: Duas redes *WiSARD*: a rede com mais nós *RAM*, responsável pela parte central (azul claro) e a rede com menos nós, responsável pela parte periférica (abóbora claro). Em verde está a janela de seleção e em vermelho está a janela de busca.

5 Conclusões

Os resultados obtidos pelas simulações foram melhores do que o esperado. O computador utilizado possui pouca capacidade de processamento, porém o rastreador executou sua tarefa em tempo real, demonstrando que o uso da rede neural sem peso *WiSARD* para o rastreamento de alvos de superfície é viável.

Os resultados das simulações descritas na seção 4.4.1 permitiram observar que quanto maior a quantidade de discriminadores da rede neural *WiSARD*, maior é o tempo médio de rastreamento *PR*, maior é a janela de busca e menor é a probabilidade de perda de um alvo rápido. Quanto menos erros de binarização ocorrem nos quadros, menos erros o rastreador comete. A introdução de um preditor simples de posição melhora a performance do rastreador sem aumentar consideravelmente o tempo médio *PR* e permite verificar se o rastreador esta cometendo erros durante a fase de detecção do alvo em um quadro. Em vídeos onde o alvo realiza manobras muito bruscas, invertendo o sentido do vetor velocidade em um espaço de tempo muito curto (movimentos impossíveis de serem realizados na prática devido a inércia dos navios), o preditor deve ser desligado para evitar erros de rastreamento.

Um método híbrido e inovador de binarização que realiza a amostragem dos *pixels* do alvo em duas faixas no modelo *YCbCr* está descrito na seção 4.4.2. Foi verificado com o resultado das simulações abordadas nesta seção que quando os *pixels* do alvo e do cenário de fundo são binarizados corretamente e aumentando a quantidade de discriminadores em torno da posição retornada pelo preditor o rastreador acompanha o alvo sem erros por mais quadros.

Considerando somente o tempo médio *PR*, nós *RAM* entre 2 e 14 *bits* de endereçamento possuem desempenho semelhante e mais eficiente que outros tamanhos de nós *RAM*, porém, nós *RAM* com 3 *bits* de endereçamento compensam mais eficientemente erros de binarização (cerca de 70% dos testes). Uma conclusão importante obtida pelos testes apresentados na seção 4.4.3 é que o tamanho dos nós *RAM* influencia a robustez do rastreador contra erros na binarização dos quadros.

Com os resultados das simulações apresentadas na seção 4.4.4 conclui-se que existe uma relação entre o posicionamento dos discriminadores em cada quadro e a robustez contra erros de binarização. Utilizando uma quantidade de discriminadores inferior e com menor espaçamento em *pixels* entre a área coberta por eles, ocorre um rastreamento sem erros por mais quadros. A configuração *GP11* foi a mais eficiente contra erros de binarização.

Para a maioria dos métodos de segmentação, excluir um *pixel PO* (*pixel* do objeto alvo da segmentação) de uma região *RG* é mais grave do que incluir um *pixel PNO* (*pixel* não pertencente ao objeto alvo da segmentação) em *RG*. Isto é observado pela forma que os métodos definem os requisitos da segmentação. Uma forma inovadora de utilização de duas redes neurais sem peso *WiSARD* em paralelo para aumentar a robustez contra erros de binarização foi apresentada na seção 4.4.5. O paralelismo se dá no nível dos discriminadores. A resposta de um discriminador pertencente à rede em paralelo é a soma das respostas de um discriminador da rede que possui nós *RAM* de 3 *bits* de endereçamento com um discriminador de outra rede que possui nós *RAM* de 15 *bits* de endereçamento. O rastreamento com duas redes neurais é mais eficiente do que com apenas uma rede neural *WiSARD*, pois consegue rastrear o alvo por mais quadros quando erros de binarização ocorrem. Desta forma, a rede neural em paralelo consegue compensar os erros de binarização gerados pelo segmentador (passos 2 e 4 do rastreador proposto). Quando

a binarização não é tão eficiente, a rede *WiSARD* é treinada com um padrão de *bits* que não representam adequadamente o modelo do alvo (passo 2). Além disto, alguns dos *pixels* da imagem de fundo são binarizados para o mesmo valor (*bit*) que os *pixels* do alvo (passo 4). As simulações mostraram que existe uma porcentagem *P* (ou intervalo de porcentagens) de *pixels* cobertos pela rede neural com nós *RAM* de 3 *bits* de endereçamento e consequentemente uma porcentagem 1-*P* cobertos pela rede neural com nós *RAM* de 15 *bits* de endereçamento que faz com que o rastreador melhore sua performance. *P* não é o mesmo para todos os vídeos. Em *VF*, *P* é de 40% a 45%. Em *VC*, *P* é de 80% a 85%. Em *VUW1*, *P* é de 20%. O valor ótimo de *P* depende de fatores como o tamanho do alvo, da janela de seleção e do contraste do alvo em relação ao fundo. O cálculo *online* de *P* pode ser feito quando o grau de confiança *C* ficar abaixo de um limiar ou quando o preditor comete um grave erro.

Para melhorar a performance do rastreador baseado na rede neural sem peso *WiSARD* podem ser investigados novas formas de distribuir os discriminadores na janela de busca, diferentes tipos de preditores, novas maneiras de separar os *pixels* cobertos pelos discriminadores das duas redes neurais (não somente com faixas horizontais), outras *features* além de *pixels* binarizados para treinar a rede neural, formas de redução da quantidade de *features* e a performance do rastreamento utilizando mais de duas redes neurais *WiSARD* em paralelo.

6 Referências

[1] Silva, N. R., "Um Algoritmo de Alarme Antecipado para Sistemas Mage Radar". Dissertação de Mestrado apresentado ao Curso de Mestrado em Engenharia Elétrica do Instituto Militar de Engenharia, 2002.
[2] Bu, Q., Sun, H., Yang, D., Zhang, J., Xie, Y., "A Video Target Tracking Method Based on Particle Filter and the Features of Affine Moment Invariants". International Conference on Mechatronics and Automation - ICMA, pp. 3275-3280, 2009.
[3] Kong, X.,Luo, Q., Zeng, G., Fan, J., "A Novel Video Object Tracking Method based on Wavelet Descriptor Matching". 15th International Conference on Intelligent Multimedia and Ambient Intelligence - IMAI, 2007.
[4] Kumar, U. P. and Shalini S., "Object Tracking In Video Sequences Based On Background Updation Using F-Tree Method". International Conference on Computational Techniques and Artificial Intelligence – ICCTAI, pp. 42-44, 2011.
[5] Available from https://www.mar.mil.br/dhn/bhmn/download/cap14.pdf.
[6] Aleksander, I., Gregorio, M., França, F. M. G., Lima, P. M. V. e Morton, H., "A brief introduction to Weightless Neural Systems". European Symposium on Artificial Neural Networks - ESANN - Advances in Computational Intelligence and Learning. pp. 299-305, 2009.
[7] França, H. L., Silva, J. C., Lengerke, O., Dutra, M. S., De Gregorio, M., França, F. M. G., "Movement Persuit Control of an Offshore Automated Platform Via a *RAM* Based Neural Network". 11thInternational Conference Control Automation Robotics & Vision – ICARCV, pp. 2437-2441, 2010.
[8] De Gregorio, M., "The Agent WiSARD Approach to Intelligent Active Video Surveillance Systems". IAPR Conference on Machine Vision Applications - MVA, pp. 331-334, 2007.
[9] Available from https://www.senado.gov.br/publicacoes/diarios/pdf/sf/2005/12/14122005/44654.pdf.
[10] Salhi, A and Jammoussi, A. Y., "Object Tracking System Using Camshift, Meanshift and Kalman Filter". World Academy of Science, Engineering and Technology, Vol. 64, pp. 674-679, 2012.
[11] Sharma, G., "Video Surveillance System: A Review". International Journal of Research in Engineering & Applied Sciences IJREAS, Vol. 2, pp. 75-85, 2012.
[12] Michele, V., Ingrid, V., Karna, B. and Paolo, B., "Unsupervised Learning of Maritime Traffic Patterns for Anomaly Detection". Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications, 9th IET , pp. 1-5, 2012.
[13] Elhabian, S. Y. and El-Sayed K. M., "Moving Object Detection in Spatial Domain Using Background Removal Techniques- State of the Art". Recent patents on computer science, Vol. 1, pp. 32-54, 2008.
[14] Konstantinova, P. , Udvarev, A., Semerdjiev, T., "A Study of a Target Tracking Algorithm Using Global Nearest Neighbor". International Conference on Computer Systems and Technologies - CompSysTech, pp. 290-295, 2003.
[15] Xiong, Y. and Chen, Y., "Study of Visual Object Tracking Technique". Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT 2009), pp. 406-409, 2009.
[16] D'Arca, E., Neil, M. R., James H., "Person Tracking Via Audio and Video Fusion". Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications, 9th IET, pp. 1-6, 2012.
[17] Sevilla-Lara, L., Learned-Miller, E., "Distribution Fields for Tracking". Computer Vision and Pattern Recognition-CVPR, IEEE Conference, pp. 1910-1917, 2012.
[18] Jia, X., Lu, H., Yang, M., "Visual Tracking Via Adaptive Structural Local Sparse Appearance Model". Computer Vision and Pattern Recognition- CVPR, IEEE Conference, pp. 1822-1829, 2012.
[19] Park, D. W., Kwon, J., and Lee , K. M., "Robust Visual Tracking Using Autoregressive Hidden Markov Model". Computer Vision and Pattern Recognition- CVPR, IEEE Conference, pp. 1964-1971, 2012.
[20] Brauy , E., Barnard, K., Palanivelu, R., "A Generative Statistical Model for Tracking Multiple Smooth Trajectories". Computer Vision and Pattern Recognition- CVPR, IEEE Conference, pp. 1137-1144, 2011.

[21] Rasekhi, J., Karami, M. and Bandarabadi, M., "Wavelet Transform and Supervised Learning Methods for Object Tracking". European Journal of Scientific Research , Vol. 41, pp. 626-631, 2010.
[22] Bouzenada, M., Batouche, M. C. and Telli, Z., "Neural Network for Object Tracking". Information Technology Journal, pp. 526-533, 2007.
[23] Janen, U., Paul, C., Wittke, M. and Hahner, J., "Multi-Object Tracking Using Feed-Forward Neural Networks". International Conference of Soft Computing and Pattern Recognition - SoCPaR, pp. 176-181, 2010.
[24] Duh, F-B and Lin., C-T, "Tracking a Maneuvering Target Using Neural Fuzzy Network". IEEE Transactions On Systems Man And Cybernetics Part B, Vol. 34, pp. 16-33, 2004.
[25] Bandeira, L. C., "NC-WiSARD: Uma Implementação sem Pesos do Modelo Neural Neocognitron". Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, UFRJ, 2010.
[26] Ludermir, T. B., Carvalho, A. P. L., Braga, A. P., De Souto, M. C. P., "Weightless Neural Models: A Review of Current and Past Works". Neural Computing Surveys 2, pp. 41-61, 1999.
[27] Aleksander, I., Thomas, W. V., Bowden, P. A., "WiSARD: a radical step forward in image recognition", Sensor Review. Vol. 4, pp. 120–124, 1984.
[28] Grieco, B. P. A., Lima, P. M. V., De Gregorio, M. and França, F. M. G., "Producing Pattern Examples From ''Mental'' Images". Neurocomputing, Vol. 73, pp. 1057–1064, 2010.
[29] Tarling, R. and Rohwer, R., "Efficient Use of Training Data in the N-tuple Recognition Method". IEEE Electronics Letters, Vol. 29, pp. 2093-2094, 1993.
[30] Gregorio, M. D., "Is That Portal Ghotic? A Hybrid System for Recognising Architectural Portal Shapes". MVA – IAPR Workshop on Machine Vision Applications, pp. 389–392, 1996.
[31] Coraggio, P., Gregorio, M. D.," WiSARD and NSP for Robot Global Localization". IWINAC(part II), pp. 449–458, 2007.
[32] Pattichis, C. S., "A Hybrid Neural Network Electromyographic System: Incorporating the WISARD Net". IEEE International Conference on Neural Networks, Vol. 6, pp. 3478-3483, 1994.
[33] Rossales, I. P. R., "Implementação em Hardware de uma Rede Neural WiSARD". Trabalho de conclusão de curso de Engenharia de Computação com Ênfase em Sistemas Embarcados apresentado à Escola de Engenharia de São Carlos, da Universidade de São Paulo, 2012.
[34] Bagheri-Golzar, S, Karami-sorkhechaghaei, F., Eftekhari-Moghadam, A., "A New Method For Video Object Tracking". The Journal of Mathematics and Computer Science, Vol. 4, pp. 120-128, 2012.
[35] Forlines, C. , Balakrishnan R., "Improving Visual Search With Image Segmentation". Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI, pp. 1093-1102, 2009.
[36] Tang, L., Tian, L. F., Steward, B. L., "Color Image Segmentation with Genetic Algorithm for In-Field Weed Sensing". Agricultural and Biosystems Engineering- ASAE, Vol. 43, No. 4, pp. 1019-1027, 2000.
[37] Marsh, M., "A literature review of image segmentation techniques and matting for the purpose of implementing Grab-cut". Available from www.cs.ru.ac.za/research/g02m1682/literature.pdf.
[38] Senthilkumaran, N. and Rajesh, R., "Edge Detection Techniques for Image Segmentation - A Survey of Soft Computing Approaches". International Journal of Recent Trends in Engineering - IJTRE, Vol. 1, No. 2, pp. 250-254, 2009.
[39] Al-Amri, S. S., Kalyankar, N. V., Khamitkar, S. D., "Image Segmentation By Using Edge Detection". International Journal on Computer Science and Engineering, Vol. 2, No. 3, pp. 804-807, 2010.
[40] Sarath, Y., "Algorithms for Image Segmentation". Thesis submitted in partial fulfillment of the requirements of BITS thesis, Birla Institute of Technology and Science, Pilani – BITS, May, 2006.
[41] Raja, S. V. K., Khadir, A. S. A., Ahamed, S. S. R., "Moving Toward Region-Based Image Segmentation Techniques: A Study". Journal of Theoretical and Applied Information Technology. Vol. 5, pp. 81-87, 2009.
[42] Chabrier, S., Laurent, H., Emile, B., "Performance evaluation of image segmentation". Application to parameters fitting. European Signal Processing Conference – EUSIPCO, 2005.
[43] Ramos, V., Muge, F., "Image Colour Segmentation by Genetic Algorithms". 11th Portuguese Conf. on Pattern Recognition, in Aurelio C. Campilho and A.M. Mendonca, pp. 125-129, 2000.
[44] Lucas, L., "Image Segmentation". Technische Universitat Munchen, 2010. Available from www.skweez.net/wp-content/uploads/2010/03/Image-Segmentation.pdf
[45] Wang, Y-H., "Tutorial: Image Segmentation". Graduate Institute Of Communication Engineering, National Taiwan University, Taipei, Taiwan, ROC.
[46] Cufí, X., Munoz, X., Freixenet, J. and Martí, J., "A Review On Image Segmentation Techniques Integrating Region And Boundary Information". Advances in Imaging and Electron Physics. Edited by Peter W. Hawkes. Academic Press. Vol. 120, pp. 1-39, 2002.
[47] Kapsalas, P., Akrivas, G., Sofou, N., "D12 Report on the object detection algorithms". Imagination, 2008.
[48] Mohanta, R. K.; Sethi B., "A Review of Genetic Algorithm application for Image Segmentation". Int. J. Computer Technology & Applications, Vol. 3, pp. 720-723, 2006.
[49] Carleer, A. P., Debeir, O., Wolff, E., "Comparison Of Very High Spatial Resolution Satellite Image Segmentations".

Proceedings Of SPIE, Vol. 5238, pp. 532-542, 2004.
[50] Thiran, J., Warscotte, V., Macq, B., "A Queued-Based Region Growing Algorithm For Accurate Segmentation Of Multi-Dimensional Digital Images". Journal Signal Processing, Vol. 60, pp. 1-10, 1997.
[51] Moutinho, A. M., "Otimização de Sistemas de Detecção de Padrões em Imagens". Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, 2011.
[52] Jepson, A. D. and Fleet, D. J., "Image Segmentation". Tutorial, 2007. Available from www.cs.toronto.edu/~jepson/csc2503/segmentation.pdf.
[53] Zhang, Y. J., "A survey on Evaluation Methods For Image Segmentation". Pattern Recognition. Vol. 29, pp. 1335-1346, 1996.
[54] Sowmya, B. and Sheelarani, B., "Colour Image Segmentation Using Soft Computing Techniques". International Journal Of Soft Computing Applications. pp. 69-80, 2009.
[55] Ma, Z., Tavares, J. M. R. S., Jorge, R. M. N., "A Review On The Current Segmentation Algorithms For Medial Images". In proceeding of: IMAGAPP 2009 - Proceedings of the First International Conference on Computer Imaging Theory and Applications, pp. 135-140, 2009.
[56] Hampton C., Persons T., Wyatt C., Zhang Y., "Survey of Image Segmentation". 1998. Available from http://csce.uark.edu/~jgauch/library/Segmentation/Hampton.1998.pdf
[57] Majeed, S. K., Saghir, M. Y., "Using Genetic Algorithm in Image Clustering". Eng. & Tech. Journal, Vol. 28, No. 13, pp. 2576-2591, 2010.
[58] St-Laurent, L., Maldague, X. and Prévost, D., "Combination Of Colour and Thermal Sensors for Enhanced Object Detection". 10th International Conference on Information Fusion - FUSION 2007, pp. 1-8, 2007.
[59] Heitz, F., Pérez, P., Bouthemy, P., "Multiscale minimization of global energy functions in some visual recovery problems". Journal Computer Vision Graphics and Image Processing - CVGIP:Image Understanding, Vol. 59, No. 1, pp. 125-134, 1994.
[60] Kanungo, P., Nanda P. K. and Samal, U. C., "Image Segmentation Using Thresholding and Genetic Algorithms". Proceedings of the Conference on Soft Computing Technique for Engineering Applications - SCT, 2006.
[61] Bawa, S., "Edge Based Region Growing". Thesis submitted in partial fulfillment of the requirements for the award of the degree of Master of Engineering in Electronics and Communication Engineering. Department of Electronics and Communication Engineering, Thapar Institute of Engineering & Technology, India, 2006.
[62] Kumar, R., Parashar, T. and Verma, G., "A Multilevel Automatic Thresholding for Image Segmentation Using Genetic Algorithm and DWT". International Journal of Electronics and Computer Science Engineering, Vol. 1, No. 1, pp. 153-160.
[63] Cho, S., Haralick, R., Yi, S., "Improvement of kittler and illingworth's minimum error thresholding". Journal Pattern Recognition, Vol. 22, pp. 609-617, 1986.
[64] Chaudhuri, D., Chaudhuri, B. B. and Murthy, C. A., "A New Split-and-Merge clustering technique". Pattern Recognition Letters, Vol. 13, pp. 399-409, 1992.
[65] Otsu, N., "A Threshold Selection Method From Gray-Level Histograms". IEEE Transactions On Systems, Man and Cybernetics, Vol. 9, No. 1, pp. 62-66, 1979.
[66] Feng, M-L., Tan, Y-P., "Contrast Adaptive Binarization Of Low Quality Document Images". IEICE Eletronics Express, Vol. 1, No. 16, pp. 501-506, 2004.
[67] Rosin, P. L., Unimodal Thresholding. Journal Pattern Recognition - PR, Vol. 34, No. 11, pp. 2083-2096, 2001.
[68] Mattos, R. S., e Veiga, Á., "Otimização de Entropia: Implementação Computacional Dos Princípios Maxent E Minxent". Scientific Electronic Library Online - Pesquisa Operacional, Vol. 22, No. 1, pp. 37-59, 2002.
[69] Zugaj, D. and Lattuati, V., "A new approach of color image segmentation based on fusing region and edge detection outputs". Journal Pattern Recognition, Vol. 31, No. 2, pp. 105-113, 1998.
[70] Canny, J., "A Computational Approach For Edge Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, issue. 6, pp. 679-698, 1986.
[71] Beucher, S., Lantuéjoul, C., "Use of Watersheds in Contour Detection". International Workshop on image processing, real-time edge and motion detection/estimation, 1979. Available from http://cmm.ensmp.fr/~beucher/publi/watershed.pdf
[72] Meyer, F., "Color Image Segmentation". In: Proceedings of 4th. International Conference on Image Processing and its Applications, pp. 303-306, 1992.
[73] Derivaux, S., Lefevre, S., Wemmert, C., Korczak, J., "On Machine Learning in Watershed Segmentation". IEEE Workshop on Machine Learning for Signal Processing, pp. 187-192, 2007.
[74] Belhomme, P., Elmoataz, A., Herlin, P., Bloyet, D., "Generalized Region Growing Operator With Optimal Scanning: Application To Segmentation Of Breast Cancer Images". In: Journal Of Microscopy, Vol. 186, n. 1, pp. 41-50, 1997.
[75] Zhang, Y.-J., "A Sumary Of Recent Progress For Segmentation Evaluation". Chapter XX, Tsinghaua University, 2006.