MODELAGEM DE OBJETOS ATRAVÉS DE RASTREAMENTO SEM PESOS

Daniel Nunes do Nascimento

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Diego Leonel Cadette Dutra
Felipe Maia Galvão França

Rio de Janeiro
Maio de 2024

MODELAGEM DE OBJETOS ATRAVÉS DE RASTREAMENTO SEM PESOS

Daniel Nunes do Nascimento

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Diego Leonel Cadette Dutra
Felipe Maia Galvão França

Aprovada por: Prof. Diego Leonel Cadette Dutra
Prof. Felipe Maia Galvão França
Prof. Claudio Miceli de Farias
Prof. Josefino Cabral Melo Lima
Prof. Alberto Ferreira de Souza

RIO DE JANEIRO, RJ – BRASIL
MAIO DE 2024

do Nascimento, Daniel Nunes

Modelagem de Objetos Através de Rastreamento sem Pesos/Daniel Nunes do Nascimento. – Rio de Janeiro: UFRJ/COPPE, 2024.

XVI, 99 p.: il.; 29, 7cm.

Orientadores: Diego Leonel Cadette Dutra
Felipe Maia Galvão França

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2024.

Referências Bibliográficas: p. 93 – 99.

1. Redes neurais sem peso. 2. WiSARD. 3. ClusWiSARD. 4. Rastreamento de objetos em vídeo. 5. Modelagem de objetos. I. Dutra, Diego Leonel Cadette *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus pais, Rosemery e José Valério.

Agradecimentos

Gostaria de agradecer a Deus, por ter chegado até aqui. Agradeço aos meus pais Rosemery e José Valério, que formaram minha base, permitindo que este momento acontecesse. Agradeço à minha esposa Amanda, companheira da minha vida, que vivenciou comigo todos os momentos complicados e angustiantes deste processo, sem a qual não teria sido possível essa conquista. Agradeço aos meus tios José Maria e Nádia, que sempre me auxiliaram nos mais diversos momentos. Um agradecimento muito grande também ao meu orientador, professor Felipe, que sempre acreditou que seria possível, mesmo quando parecia impossível concluir esta tese. Agradeço ao professor Diego, que foi fundamental para que esta tese se concretizasse. Agradeço aos meus amigos Douglas e Kleber, sempre presentes nesta trajetória acadêmica e fundamentais para que eu pudesse chegar até aqui. Agradeço também minha irmã Luiza, meus primos, tios, parentes e amigos que sempre estiveram comigo. Agradeço a todos que de alguma forma contribuíram neste caminho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

MODELAGEM DE OBJETOS ATRAVÉS DE RASTREAMENTO SEM PESOS

Daniel Nunes do Nascimento

Maio/2024

Orientadores: Diego Leonel Cadette Dutra
Felipe Maia Galvão França

Programa: Engenharia de Sistemas e Computação

Esta tese apresenta um método para realizar a criação de modelos de objetos rastreados em frames de vídeo. Na abordagem proposta, tanto o rastreamento quanto a modelagem dos objetos são realizados em tempo real, sem utilização de nenhum tipo de conhecimento prévio sobre os objetos de interesse a serem modelados. O objetivo principal deste trabalho é gerar modelos que forneçam um entendimento sobre as estruturas visuais dos objetos observados, mapeando possíveis transições entre os aspectos aprendidos. As únicas informações utilizadas como entradas para o sistema desenvolvido são as coordenadas da localização do alvo no primeiro frame, e a partir deste momento, o rastreamento é executado para encontrar as localizações corretas em cada um dos frames observados. As respostas de localização obtidas em cada um dos frames são passadas para o modelador de objetos, responsável por mapear os aspectos e as possíveis transições entre aspectos. Os modelos criados possuem representações visuais que podem ser utilizadas para determinar caminhos de estados entre aspectos de um mesmo objeto, ou até mesmo visualizar partes de um objeto que estejam sofrendo oclusão parcial. O desenvolvimento do sistema proposto nesta tese foi baseado em redes neurais sem peso, utilizando os modelos WiSARD, ClusWiSARD e DRASiW.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

OBJECT MODELING THROUGH WEIGHTLESS TRACKING

Daniel Nunes do Nascimento

May/2024

Advisors: Diego Leonel Cadette Dutra
Felipe Maia Galvão França

Department: Systems Engineering and Computer Science

This thesis presents a method for creating models of objects being tracked in video frames. In the proposed approach, both object tracking and modeling are performed in real time, without using any type of prior knowledge about the objects to be modeled. The main purpose of this work is to generate models that provide an understanding of the visual structures of the objects being observed, mapping possible transitions between the learned aspects. The only information used as input to the developed system are the target location coordinates in the first frame, and from this moment on, tracking is performed to find the correct locations in each of the frames. The location coordinates responses obtained in each of the frames are passed to the object modeler, which is responsible for mapping the aspects and possible transitions between aspects. The created models have visual representations that can be used to determine state paths between aspects of the same object, or even visualize occluded parts of an object. The development of the system proposed in this thesis was based on weightless neural networks, using the models WiSARD, ClusWiSARD and DRASiW.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Um dos grandes desafios da visão computacional é o de representar objetos do mundo real através de modelos compreendidos por sistemas computacionais. Neste tipo de problema, busca-se transformar as informações visuais em representações digitais, que sejam processáveis e utilizáveis por diversos tipos de aplicação. Além de mapear as informações visuais dos objetos, um grande desafio consiste em obter representações significativas acerca dos aspectos dos objetos, ou seja, representações que forneçam algum entendimento sobre as estruturas dos objetos observados. Para um ser humano pode ser muito simples olhar para um objeto e conseguir imaginar como seriam as partes que não podem ser visualizadas no momento, porém, esta tarefa não é trivial para ser executada por sistemas automatizados. Neste contexto, esta tese busca executar em tempo real, o rastreamento de objetos em vídeos e, simultaneamente, construir representações que permitam um entendimento sobre os aspectos visuais apresentados pelos objetos.

1.1 Rastreamento de objetos

O problema do rastreamento de objetos consiste em identificar corretamente a localização dos objetos de interesse em todos os frames de um determinado vídeo. Esta tarefa possui bastante relevância, podendo ser utilizada em diferentes tipos de aplicação, como por exemplo, aplicações reconhecimento de gestos [1], navegação autônoma de veículos [2] ou sistemas de monitoramento de segurança [3]. Para realizar o rastreamento de objetos em vídeo, utilizam-se técnicas de processamento de imagens e reconhecimento de padrões, a fim de contornar os complexos desafios envolvidos, dentre os quais se destacam a ocorrência de mudanças nos aspectos apresentados pelos objetos, mudanças de escala acarretadas por afastamentos ou aproximações do alvo em relação ao ponto de observação, problemas envolvendo a luminosidade do ambiente, problemas envolvendo oclusão, onde o objeto perseguido se encontra parcialmente ou totalmente escondido por outros objetos presentes na

cena observada, ocorrência de objetos que se deslocam com muita velocidade ou presença de imagens desfocadas. As imagens a seguir ilustram algumas destas situações que devem ser contornados neste tipo de problema.

Figura 1.1: Mudanças nos aspectos. Um mesmo objeto em movimento pode variar as formas apresentadas, dificultando sistemas de rastreamento.

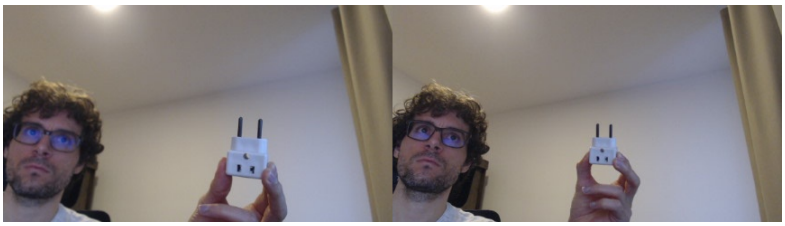

Figura 1.2: Mudanças de escala. Um mesmo objeto pode se aproximar ou afastar em relação ao ponto de observação, modificando sua aparência e dificultando o seu reconhecimento.

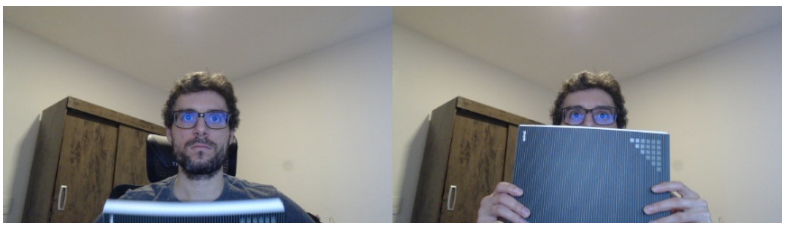

Figura 1.3: Oclusão. Um objeto rastreado pode sofrer oclusão parcial ou total, impossibilitando um reconhecimento preciso.

O rastreamento de objetos é uma parte relevante para a abordagem do problema de representação de objetos tratado nesta tese, onde o rastreador de objetos empregado como parte do sistema proposto não utiliza informações prévias sobre os aspectos dos objetos e fornece as localizações dos alvos para o modelador de objetos, que cria os modelos de maneira totalmente online.

1.2 Representação de objetos

A conversão de objetos do mundo real para representações compreensíveis por sistemas computacionais é um grande desafio, pois além da obtenção das representações digitais dos objetos, ainda se faz necessário criar sistemas com capacidade de obter algum tipo de conhecimento sobre os objetos observados. O entendimento sobre as estruturas dos objetos são de especial interesse para aplicações de robótica que manipulam os objetos com o objetivo de movê-los de um estado inicial até um estado objetivo [4, 5].

Existem algumas formas de se representar os objetos, como por exemplo, através de modelos formados por voxels [6, 7], que são estruturas volumétricas representando as menores unidades de uma imagem em 3D, de maneira semelhante aos pixels, que são as menores unidades que formam uma imagem em 2D; através de nuvens de pontos [8, 9], que são conjuntos de pontos com suas coordenadas definidas no espaço tridimensional, representando a superfície de um objeto; ou utilizando múltiplas imagens em 2D, obtidas através de diferentes pontos de visualização de um mesmo objeto [10, 11]. As figuras a seguir exemplificam algumas formas de representação de objetos.

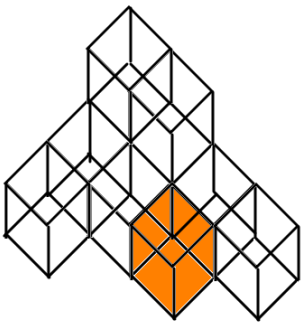

Figura 1.4: Representação de um objeto formado por voxels. Um voxel é possui o formato de um cubo e é menor unidade formadora de uma imagem em 3D, de maneira semelhante a um pixel em uma imagem 2D.

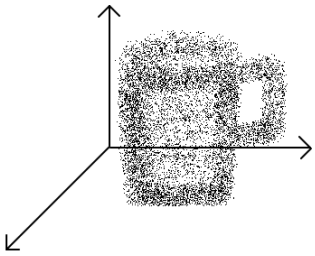

Figura 1.5: Representação de um objeto formado por núvens de pontos. São pontos com coordenadas (x, y, z), obtidos da superfície do objeto através de sensores de captura de profundidade.

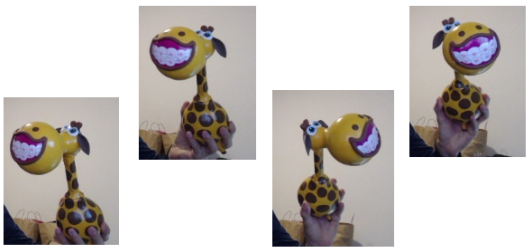

Figura 1.6: Representação de um objeto através de múltiplas visões. Aspectos obtidos através de diferentes pontos de observação fornecem uma representação para um objeto.

A obtenção de dados para mapear os objetos do mundo real pode ser feita através de sistemas de câmeras de visão estéreo, onde as imagens de um objeto são obtidas por câmeras posicionadas em diferentes pontos de observação para determinar as localizações dos pixels no espaço, reconstruindo os objetos [12, 13], ou até mesmo pode-se utilizar câmeras RGB-D, que capturam imagens com informações de profundidade, permitindo um mapeamento em 3D para os aspectos dos objetos. [14, 15]. O presente trabalho utiliza imagens obtidas a partir de uma única câmera para mapear um objeto de interesse e construir um modelo representativo através de diversos aspectos aprendidos pelo sistema. A inovação dos modelos de objetos apresentados nesta tese consiste no método de representação através do conhecimento armazenado no modelo de aprendizado WiSARD, descrito na Seção 2.1, gerando informações que permitem que o sistema apresentado possua algum tipo de entendimento sobre as estruturas visuais dos objetos, sendo possível visualizar aspectos aprendidos, assim como as relações entre aspectos mapeadas nos modelos.

1.3 Trabalhos relacionados

Esta seção apresenta alguns dos trabalhos relacionados aos assuntos abordados nesta tese, dando destaque para trabalhos desenvolvidos com redes neurais sem peso relacionados ao problema de rastreamento de objetos e apresentando alguns outros importantes rastreadores encontrados na literatura.

1.3.1 Trabalhos envolvendo redes neurais sem peso

A seguir, são apresentados alguns trabalhos desenvolvidos com redes neurais sem peso, correlatos com o problema de rastreamento de objetos.

- **Movement persuit control of an offshore automated platform via a RAM-based neural network**

  Neste trabalho, França *et al.* utilizaram o modelo de redes neurais sem peso WiSARD para acompanhar a cadência de navios, identificando e seguindo pontos de interesse que podem definir um modelo do movimento da embarcação observada. Este trabalho executa uma busca ao redor da imagem em baixa resolução, buscando os pontos com as melhores pontuações retornadas pelo modelo treinado, e após serem identificados, são verificados novamente na imagem com a resolução original. Essa abordagem de se utilizar a WiSARD para buscar o alvo de interesse em imagens de baixa resolução foi utilizada nesta tese para o desenvolvimento do módulo de detecção de objetos, apresentado no Capítulo 4, responsável por localizar o alvo em escalas diversas.

- **Online tracking of multiple objects using WiSARD**

  O modelo de redes neurais sem peso WiSARD foi aplicado ao problema de rastreamento de objetos por De Carvalho *et al.* [16], realizando retreinos em tempo real, para lidar com problemas como mudanças nos formatos apresentados pelos objetos ou ruídos no background das imagens. Este trabalho realiza uma busca local, para encontrar a localização do alvo. A partir desta aplicação do modelo WiSARD ao problema de rastreamento, outros trabalhos foram desenvolvidos, realizando o rastreamento com múltiplos discriminadores WiSARD [17], [18], que posteriormente, resultaram no ponto de partida para desenvolvimento desta tese.

- **WiSARD rp for change detection in video sequences**

  Detecções de mudanças no background (fundo da imagem) podem ser úteis para auxiliar sistemas de rastreamento de objetos, e o trabalho desenvolvido por De Gregorio e Giordano [19] utiliza redes neurais sem peso para detectar essas mudanças. O mecanismo de treinamento atualiza continuamente um

modelo do background, baseado nos pixels da imagem, aprendendo mudanças ocorridas nas cores dos pixels ao longo do tempo. O modelo aprendido é utilizado para classificar os pixels entre pertencentes ao background ou pertencentes ao foreground (partes da imagem que não pertencem ao fundo, como por exemplo, objetos se movimentando).

- **Sistema de rastreamento visual de objetos baseado em movimentos oculares sacádicos**

  No trabalho apresentado por Andrade [20], o rastreamento de objetos é abordado através de uma solução biologicamente inspirada nos movimentos sacádicos dos olhos. Esta arquitetura foi baseada no modelo de redes neurais sem peso VG-RAM (Virtual Generalizing Random Access Memory) [21] e realiza uma busca visual de pontos de interesse previamente treinados.

1.3.2 Trabalhos relacionados com rastreamento de objetos

Os trabalhos listados a seguir apresentam algumas das abordagens utilizadas para tratar do problema de rastreamento de objetos em vídeo.

- **Forward-backward error: Automatic detection of tracking failures**

  Kalal, Mikolajczyk e Matas [22] apresentaram o rastreador Median Flow, baseado no rastreador Lucas-Kanade [23], que realiza o rastreamento de objetos para frente e para trás nos frames dos vídeos, e com base no erro obtido através da diferença entre as trajetórias, identifica as localizações dos objetos rastreados.

- **Tracking-learning-detection**

  O TLD - Tracking-learning-detection, também desenvolvido por Kalal, Mikolajczyk e Matas [24], é um rastreador baseado no rastreador Median Flow, e utiliza uma separação em módulos, um para executar o rastreamento, um responsável pelo aprendizado e um para realizar a detecção. O rastreador segue o objeto em cada frame, o detector localiza as aparências que foram vistas para corrigir o rastreador e o módulo de aprendizagem estima erros executados pelo detector e o atualiza para evitar que os erros sejam propagados.

- **Visual tracking with online multiple instance learning**

  Babenko, Yang e Belongie desenvolveram o MIL Tracker - Multiple Instance Learning Tracker [25], [26], um rastreador de objetos baseado no aprendizado de múltiplas instâncias, onde o classificador é treinado a partir de conjuntos de exemplos agrupados, ao invés de rotular cada instância individualmente. O

objetivo do classificador é conseguir diferenciar fragmentos positivos (classes de objetos) e fragmentos negativos (partes da imagem pertencentes ao background). Se um conjunto de exemplos for rotulado positivamente, é esperado que pelo menos uma instância positiva esteja presente neste grupo. Desta forma, um conjunto positivo pode conter alguns bounding boxes delimitando possíveis objetos, sendo necessária a utilização do algoritmo de aprendizado para determinar qual instância presente em cada grupo classificado como positivo representaria uma melhor resposta.

- **Real-time tracking via on-line boosting**

  Grabner, Grabner, e Bischof [27] propuseram um algoritmo de seleção de features para rastreamento online utilizando o algoritmo AdaBoost [28]. É um rastreador que se adapta durante o rastreamento e dependendo do tipo de background nas cenas de rastreamento, o algoritmo seleciona as features que melhor se aplicam a cada tipo de background. O sistema utiliza features como transformada de Haar, histogramas e padrões binários.

- **High-speed tracking with kernelized correlation filters**

  Henriques *et al.* [29] desenvolveram um classificador treinado com conjuntos de imagens contendo regiões de redundâncias onde aplica-se filtros de correlação a estas áreas para prever futuras posições dos objetos rastreados. Este trabalho realiza a implementação de um rastreador aplicando kernel Gaussiano em descritores HOG (Histogram of Oriented Gradients).

- **Visual object tracking using adaptive correlation filters**

  O trabalho desenvolvido por Bolme *et al.* [30] realiza o rastreamento de objetos utilizando o filtro de correlação Minimum Output Sum of Squared Error (MOSSE). A correlação é calculada no domínio de Fourier, e o objetivo é minimizar a soma dos erros quadráticos entre a saída real e a saída prevista da convolução.

- **Learning to track at 100 FPS with deep regression networks**

  O GOTURN - Generic Object Tracking Using Regression Networks, apresentado por Held, Thrun e Savarese [31] utiliza redes neurais convolucionais profundas para realizar o treinamento de objetos de maneira offline, obtendo um modelo robusto para ser utilizado no rastreamento online dos objetos.

- **Discriminative correlation filter with channel and spatial reliability**

  Neste trabalho, Alan *et al.* [32] introduzem os conceitos de confiabilidade espacial e de canal, aprimorando o rastreamento por filtros de correlação discriminativa. Neste rastreador, o mapa de confiabilidade espacial ajusta o filtro

à parte adequada do objeto, permitindo realizar a busca em uma região ampliada, além de possibilitar o rastreamento de objetos não retangulares. As pontuações de confiabilidade de canal são utilizadas para avaliar a qualidade dos filtros aprendidos que são utilizados para determinar a localização.

1.4 Proposta da tese

Dentro do contexto de rastreamento e modelagem de objetos, esta tese se propõe a criar um método para realizar as duas tarefas simultaneamente, gerando modelos descritivos dos objetos rastreados, fornecendo um entendimento sobre os aspectos observados em conjunto com possíveis transições entre os aspectos. Neste trabalho, ambas as tarefas de rastrear e modelar partem do conhecimento zero, ou seja, nenhum tipo de treinamento prévio é realizado e todo o aprendizado é executado a partir do primeiro frame de cada vídeo observado. As coordenadas de localização dos objetos no primeiro frame são as únicas informações passadas para o sistema.

Para realizar a criação dos modelos, o rastreador identifica as localizações dos alvos perseguidos, passando-as para o modelador de objetos, que aprende as representações dos aspectos, assim como as possíveis transições entre os aspectos dos objetos. Os modelos criados possuem representações visuais, de maneira que é possível utilizá-los em aplicações para visualizar partes escondidas de um objeto ou determinar caminhos de transições entre aspectos para movimentar um objeto de estado inicial até um estado final. A Figura 1.7 apresenta de maneira resumida, a proposta de trabalho desta tese, onde um objeto desconhecido é rastreado e modelado em tempo real resultando em um sistema que possui conhecimento sobre os aspectos e relações entre aspectos dos objetos observados.

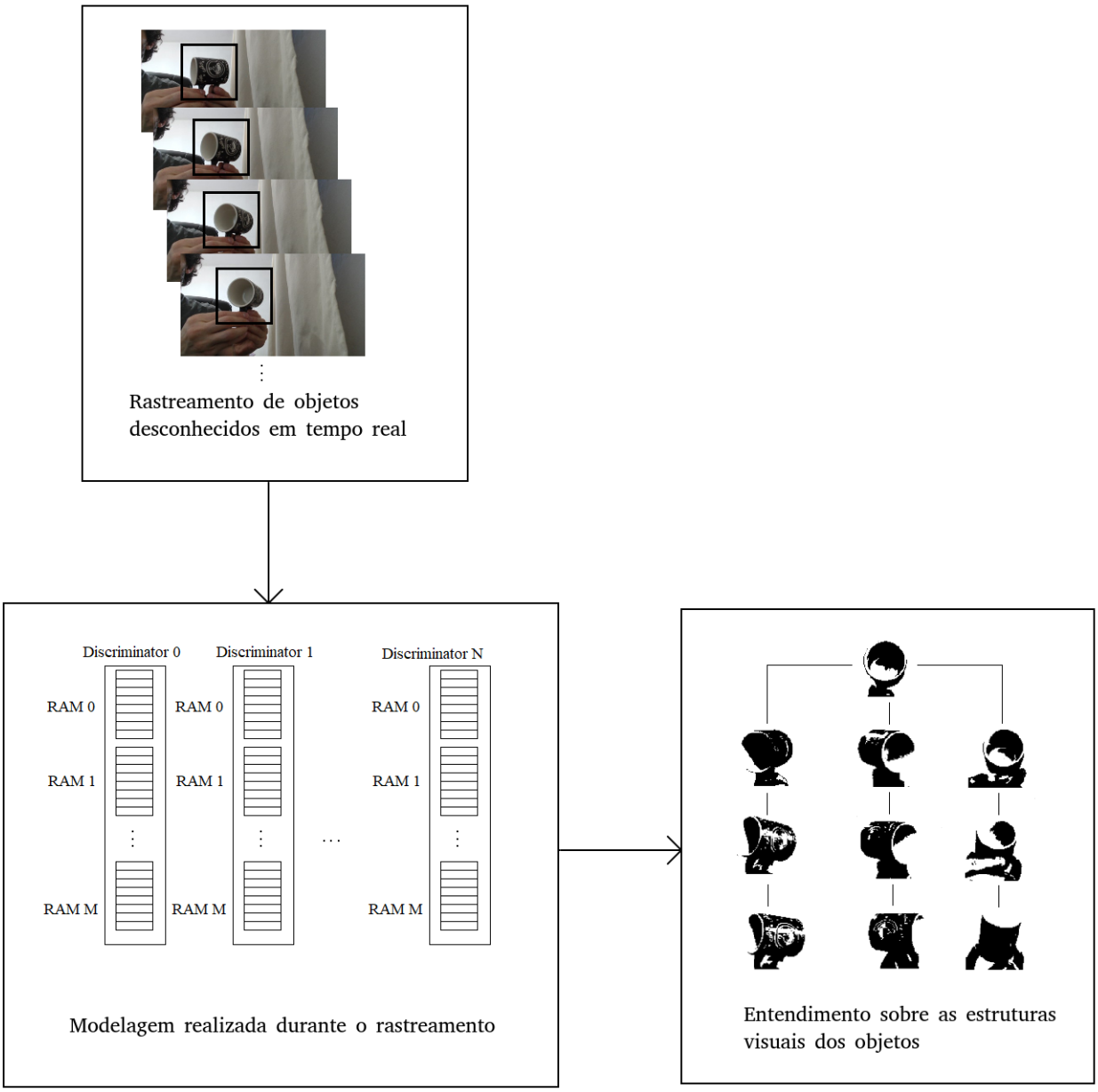

Figura 1.7: Modelagem em tempo real utilizando redes neurais sem peso. Um objeto desconhecido é rastreado e modelado a partir da sua localização informada no primeiro frame do vídeo.

1.5 Organização da tese

A organização desta tese segue a seguinte estrutura: O Capítulo 2 apresenta uma revisão sobre os modelos de redes neurais sem peso WiSARD, AutoWiSARD, ClusWiSARD e DRASiW, que são os modelos base para o desenvolvimento deste trabalho; Capítulo 3 descreve o rastreador de objetos baseado em redes neurais sem peso, que foi o ponto de partida para o desenvolvimento desta tese; o Capítulo 4, apresenta as evoluções realizadas neste rastreador, que são contribuições desta tese; o Capítulo 5 apresenta a principal contribuição desta tese, que é a criação de modelos em tempo real de objetos rastreados a partir de imagens de vídeo; no Capítulo 6 se encontram os experimentos realizados e os resultados obtidos e finalmente, o Capítulo 7 contém as considerações finais e possíveis trabalhos futuros. As principais contribuições desta tese foram publicadas primeiramente na revista Neural Computing and Applications, pela editora Springer Nature, no artigo Object modeling through weightless tracking [33].

Capítulo 2

Redes neurais sem peso

As redes neurais tradicionais [34], são modelos de aprendizado inspirados no funcionamento dos neurônios biológicos, onde o conhecimento adquirido fica armazenado nos pesos sinápticos que conectam os neurônios da rede e que são atualizados durante a etapa de treinamento. Neste tipo de modelo, cada valor de entrada de um neurônio é multiplicado pelo seu respectivo peso sináptico, e o somatório dessas multiplicações é aplicado como entrada de uma função que resulta no valor de saída do neurônio. A organização desses neurônios em camadas [35] é amplamente utilizada, onde a saída de um neurônio é aplicada como entrada de todos os neurônios da camada seguinte e assim sucessivamente até a saída retornada pela rede neural.

Por outro lado, nas redes neurais sem peso, os neurônios são definidos através de memórias RAM (Random Access Memory) [36] que armazenam o aprendizado nos endereços de memória acionados durante o treinamento. As redes neurais sem peso, amplamente estudadas, têm sido aplicadas nos mais variados tipos de problema, como por exemplo, processamento de linguagem natural [37], processamento de áudio [38], detecção precoce de crises epilépticas [39] e classificação de trajetórias de GPS [40]. As pesquisas envolvendo redes neurais sem peso continuam sendo bastante exploradas, resultando em trabalhos como [41–45].

A seguir, o modelo WiSARD [46] é apresentado em conjunto com suas variações AutoWiSARD [47], ClusWiSARD [48, 49] e DRASiW [50], que são as redes neurais sem peso utilizadas como base para o desenvolvimento desta tese. Existem ainda diversos outros modelos de redes neurais sem peso como PLN [51], GSN [52], GRAM [53], VG-RAM [21], GNU [54] e SDM [55].

2.1 WiSARD

O modelo de rede neural sem peso WiSARD (Wilkie, Stonham and Aleksander's Recognition Device) [46] é um classificador que utiliza uma estrutura de discriminadores para representar cada classe de um determinado problema. Cada discriminador é

formado por memórias RAM, que são responsáveis por armazenar o conhecimento obtido durante a etapa de aprendizado. Para realizar a classificação de novos padrões, estes são apresentados aos discriminadores treinados a fim de verificar o nível de similaridade com cada uma das classes do problema.

2.1.1 Treinamento

No treinamento do modelo WiSARD, cada padrão de entrada possui uma classe rotulada e deve ser apresentado ao discriminador de classe correspondente. Os padrões de entrada devem ser mapeados em endereços a serem ativados nas memórias RAM dos discriminadores. Cada exemplo de treinamento é convertido em um conjunto de $\boldsymbol{n}$-tuplas de endereços que são utilizados como entradas para as $\boldsymbol{N}$ memórias RAM do discriminador de classe correspondente, indicando os endereços que serão ativados.

A Figura 2.1 ilustra o treinamento de um discriminador formado por 5 RAMs, cada uma com 3 bits de endereçamento ($\boldsymbol{n = 3}$ e $\boldsymbol{N = 5}$). Sendo $n = 3$, o padrão de entrada é convertido em grupos de 3 bits, cada um representando um endereço de memória a ser ativado em uma das 5 RAMs presentes no discriminador. Neste caso, como o exemplo de treinamento é uma imagem, cada conjunto de 3 pixels é convertido em um endereço de 3 bits, onde o primeiro endereço formado indica o endereço a ser ativado na primeira RAM, o segundo endereço formado, indica o endereço a ser ativado na segunta RAM e assim sucessivamente para todos os endereços de 3 bits formados a partir dos pixels da imagem. A seleção dos pixels para a formação dos endereços segue uma ordenação aleatória, que deve ser seguida por todos os padrões de entrada apresentados para o discriminador. A numeração presente em cada pixel da imagem fornece a ordem de seleção do pixel, e a cor do pixel representa o bit utilizado na formação do endereço, sendo 0 para pixel branco e 1 para pixel preto.

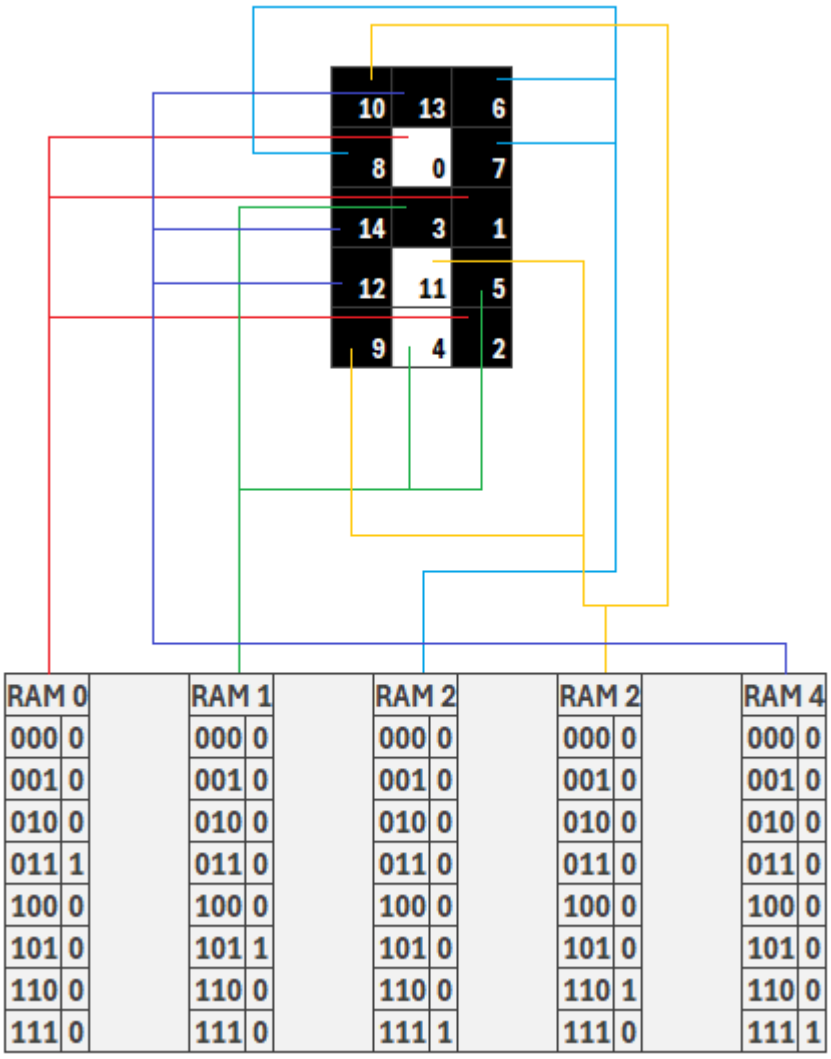

Figura 2.1: Treinamento de um discriminador. Os pixels da imagem são convertidos em endereços de memória para serem ativados no discriminador. O primeiro conjunto de 3 pixels forma o endereço da primeira RAM, o segundo conjunto de 3 pixels forma o endereço da segunda RAM e assim sucessivamente.

2.1.2 Classificação

A classificação de um padrão desconhecido segue o mesmo mapeamento realizado durante o treinamento. Desta forma, os pixels do novo padrão são mapeados para formar endereços que devem ser apresentados para o discriminador treinado, a fim de verificar a quantidade de RAMs que foi ativada. Caso o endereço formado pelo novo padrão tenha sido marcado durante o treinamento na RAM correspondente, esta RAM retorna o valor 1 e caso a RAM não tenha sido ativada durante o treinamento, o valor de retorno é 0. A obtenção de uma resposta para a classe do padrão de entrada é obtida apresentando-se o padrão para os discriminadores de cada uma das classes, sendo aquele de maior pontuação, o selecionado para retornar a resposta.

A Figura 2.2 apresenta a obtenção da taxa de ativação de um padrão classificado por um discriminador treinado, onde 3 das 5 RAMs foram ativadas, obtendo-se uma pontuação de ativação igual a 0,6. A Figura 2.3 ilustra a classificação de um padrão desconhecido através de sua apresentação para todos os discriminadores treinados, sendo aquele com a maior ativação, o escolhido para retornar a resposta de classificação.

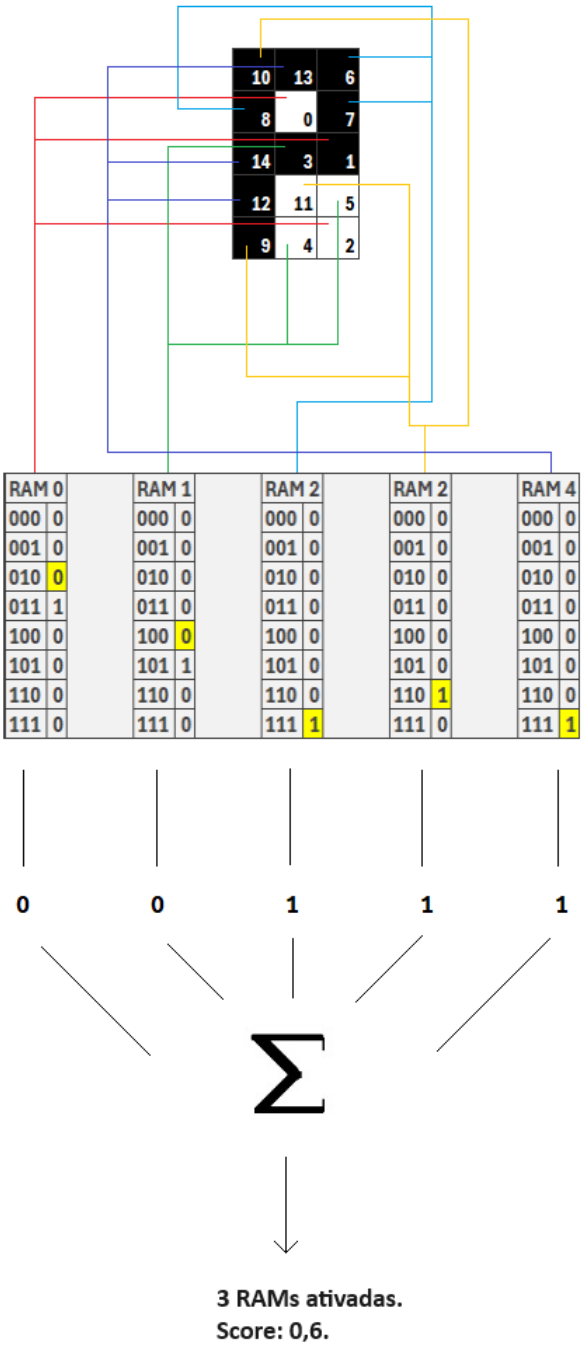

Figura 2.2: Classificação de um padrão por um discriminador. O mapeamento aleatório de pixels deve ser o mesmo utilizado no treinamento. Para cada RAM, se o endereço selecionado estiver marcado, esta RAM foi ativada. A pontuação retornada pelo discriminador é a porcentagem de RAMs ativadas e representa o nível de similaridade do padrão apresentado com a classe representada pelo discriminador.

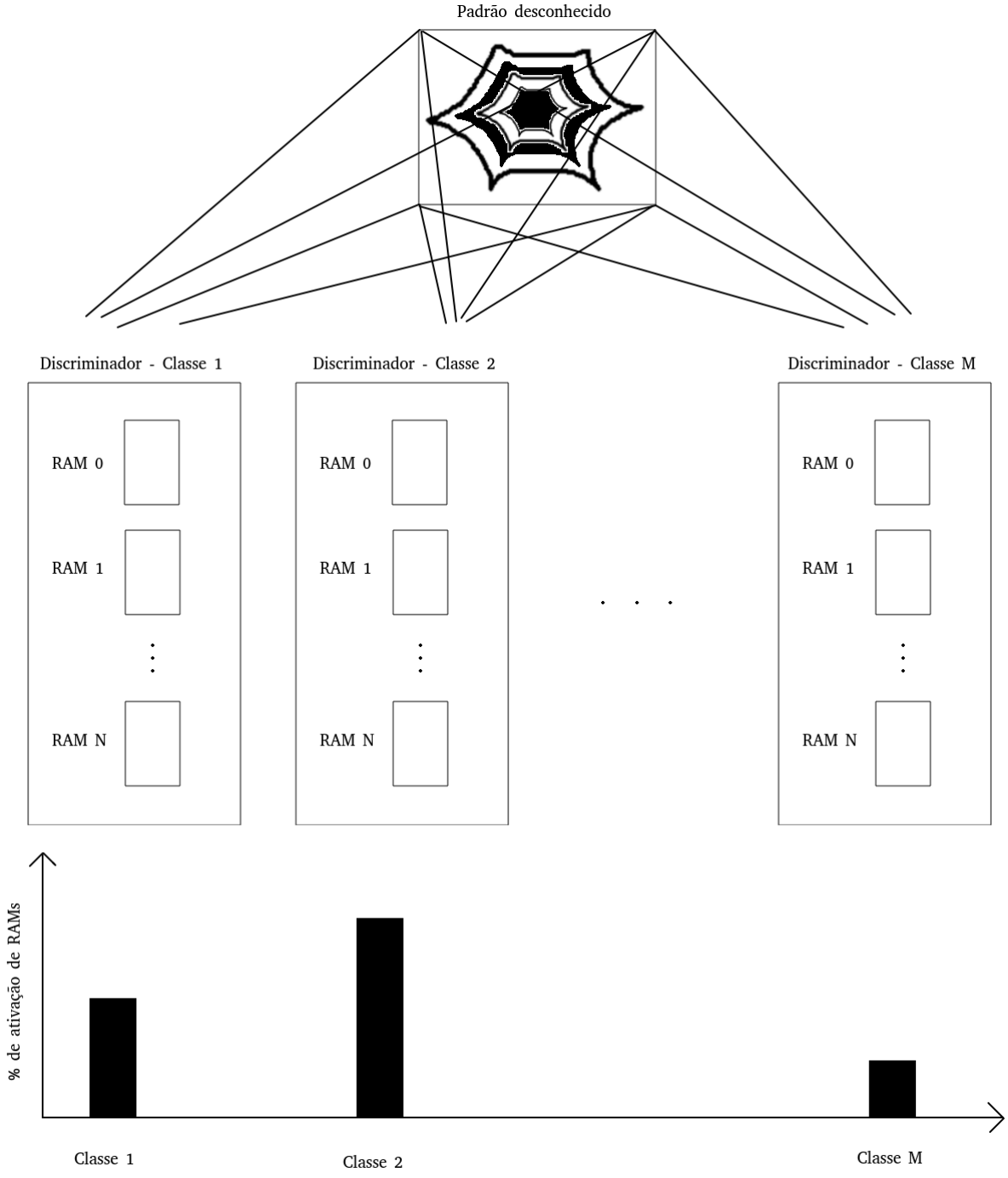

Figura 2.3: Determinação da classe de um padrão desconhecido. O padrão é apresentado para cada um dos discriminadores treinados, sendo aquele com a maior taxa de ativação de RAMs o escolhido para representar a classe do padrão desconhecido.

2.2 AutoWiSARD

A AutoWiSARD[47] é um modelo utilizado para determinar se existe a necessidade de criar um novo discriminador para representar determinada classe ou se um discriminador já existente deve ser retreinado. Sendo assim, um novo exemplo apresentado ao sistema é avaliado por todos os discriminadores presentes até o momento, e caso a resposta do melhor discriminador esteja abaixo do primeiro limiar, um novo discriminador é criado; caso a resposta do melhor discriminador esteja acima do segundo limiar, nada é feito, pois assume-se que já existe outro discriminador capaz de representar o exemplo de treinamento; e por fim, caso a resposta retornada pelo melhor discriminador esteja entre os dois limiares, o treinamento parcial com esta nova instância apresentada é realizado com probabilidade p, e a criação de um novo discriminador se dá com uma probabilidade $1-p$, onde $p = (r_{best} - w_{min})/(w_{max} - w_{min})$, com r_{best} sendo a pontuação obtida pelo melhor discriminador, e $wmin$ e $wmax$ sendo os limiares que determinam o intervalo da janela de aprendizado. O treinamento parcial consiste em realizar o treinamento de uma quantidade suficiente de neurônios para que o discriminador selecionado retorne uma pontuação $r_{best} = w_{max}$.

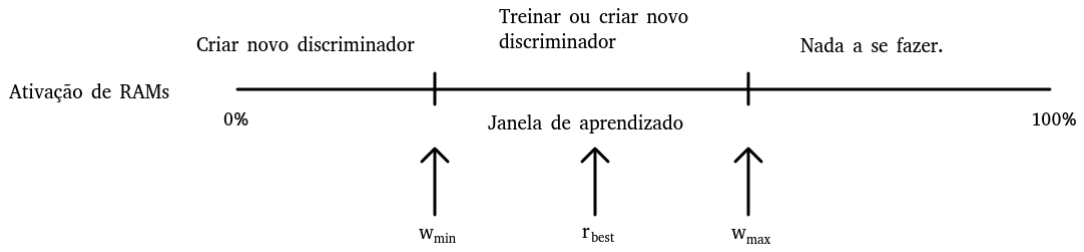

Figura 2.4: Janela de aprendizado segundo o modelo AutoWiSARD.

2.3 ClusWiSARD

O modelo ClusWiSARD [48, 49], é uma variação da WiSARD utilizada para problemas de clusterização. Dentro de uma mesma classe de um problema, os padrões de treinamento podem apresentar características muito diferentes, o que pode acarretar em discriminadores que generalizam demais, podendo ocasionar classificações errôneas. Para solucionar este problema, a ClusWiSARD define clusters de padrões dentro de uma mesma classe de treinamento, onde cada grupo é representado por um único discriminador.

Para que um cluster absorva um exemplo de treinamento, este deve pertencer à mesma classe do discriminador, e a pontuação retornada pelo discriminador para a classificação deste exemplo deve ser maior ou igual a $\boldsymbol{min(1, s + size(d)/\gamma)}$, com $\boldsymbol{s}$ sendo uma pontuação de similaridade mínima previamente determinada, $\boldsymbol{size(d)}$ representando a quantidade de padrões treinados no discriminador até o momento, e $\boldsymbol{\gamma}$ sendo um limiar previamente determinado para o intervalo de crescimento, onde quanto maior este valor, maior a quantidade de padrões que podem ser absorvidos por um discriminador. Neste trabalho, a ClusWiSARD foi utilizada para realizar o aprendizado dos aspectos que formam o modelo de um objeto e esta utilização é apresentada no Capítulo 5. O algoritmo original da ClusWiSARD é mostrado a seguir.

Algoritmo 1 Algoritmo ClusWiSARD

Entrada: s = pontuação de similaridade mínima
Entrada: γ = intervalo de crescimento
  para cada padrão de entrada i pertencente ao conjunto de treinamento **faça**
    para cada discriminador d presente na ClusWiSARD **faça**
      se $classe(i) = classe(d)$ & $score(d, i) \geq min(1, s + size(d)/\gamma)$ **então**
        Discriminador d aprende i
        $size(d) \leftarrow size(d) + 1$
      fim se
    fim para
    se Nenhum discriminador aprendeu i **então**
      Um novo discriminador d' é criado
      d' é adicionado ao conjunto de discriminadores da ClusWiSARD
      $classe(d') \leftarrow classe(i)$
      d' aprende i
      $size(d') \leftarrow 1$
    fim se
  fim para

2.4 DRASiW

O modelo DRASiW [50] é uma variação do modelo WiSARD, que permite seguir o caminho inverso ao realizado no treinamento dos discriminadores, de modo que, é possível obter uma representação visual do conhecimento armazenado em cada um dos discriminadores treinados. Essas representações visuais são chamadas de

imagens mentais, e são obtidas convertendo-se os endereços marcados nas memórias RAM em pixels a serem designados para formar uma imagem. Esta conversão de endereços em valores de pixels deve seguir a mesma ordenação de seleção aleatória de pixels utilizada no treinamento do discriminador. Para realizar a montagem de uma imagem mental, pode-se selecionar os endereços com maiores quantidades de marcações em cada uma das RAMs para gerar os valores para os pixels da imagem mental. A Figura 2.5 ilustra os passos realizados para criar uma imagem mental representante do conhecimento armazenado em um discriminador treinado. Neste exemplo, o endereço marcado na RAM 0 é o endereço 001, que convertido em cores de pixels, seria correspondente à sequência de cores branco, branco e preto a serem atribuídas aos pixels de ordem de seleção 0, 1 e 2. Seguindo para a RAM 1, o endereço 011 é convertido na sequência de cores branco, preto e preto a serem atribuídas para os pixels de ordem de seleção 3, 4 e 5. Este processo se repete até finalizar o mapeamento de todas as RAMs em pixels para formar uma imagem.

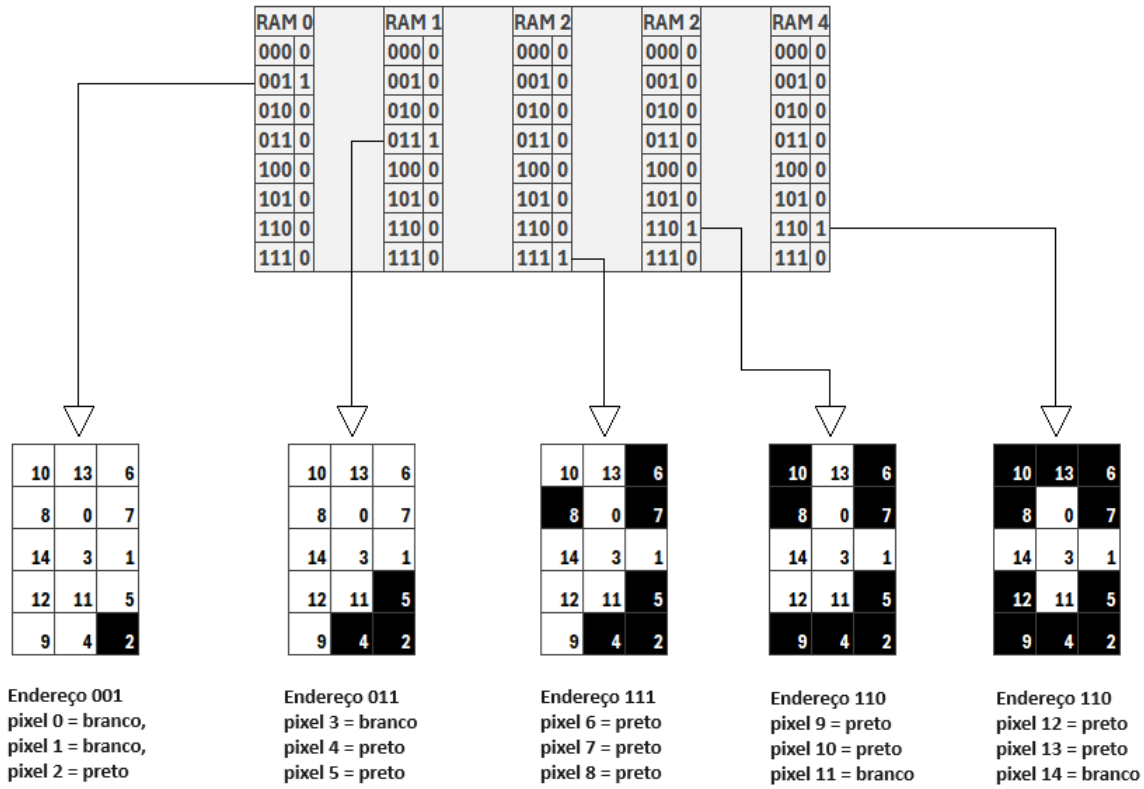

Figura 2.5: Processo de construção de imagem mental a partir de um discriminador treinado. Os endereços com maior ativação em cada uma das RAMs são selecionados para montar os pixels da imagem mental transformando os bits dos endereços em cores para os pixels, seguindo a mesma ordem aleatória de seleção de pixels utilizada no treinamento.

Neste trabalho, as imagens mentais obtidas a partir dos discriminadores treinados são utilizadas para visualizar os aspectos dos objetos modelados. Além da obtenção de imagens mentais para representar os objetos, os modelos aprendidos possuem as transições entre os aspectos, possibilitando um entendimento sobre as relações entre os diferentes pontos de vista de um mesmo objeto. Estes modelos formados a partir de imagens mentais, apresentados nesta tese, são chamados de modelos mentais e o processo de criação é apresentado no Capítulo 5.

Capítulo 3

Rastreador de objetos sem pesos

Este capítulo descreve a base inicial para o desenvolvimento deste trabalho, que foi o rastreador de objetos baseado em redes neurais sem peso [17, 18]. Nesta primeira versão, a solução encontrada para rastrear um objeto que pode se apresentar de diversas maneiras foi inspirada na memória humana, onde o conhecimento aprendido pode ser armazenado em memórias de curta duração ou longa duração [56]. Neste rastreador, essas memórias de curto e longo prazos são representadas através de discriminadores WiSARD que podem ser acessados e descartados rapidamente, funcionando como se fossem uma memória de curto prazo ou podem ficar armazenados durante um longo período do rastreamento, funcionando como memórias de longo prazo.

O modelo de rastreamento em vídeo baseado em discriminadores WiSARD possui um bom desempenho, executando todas as tarefas em tempo real, sem necessidade de nenhum tipo de treinamento prévio. Sendo assim, foi utilizado nesta pesquisa como ponto de partida, com o objetivo de desenvolver melhorias, assim como utilizá-lo como parte do sistema modelador de objetos em tempo real. As próximas seções descrevem o funcionamento deste modelo de rastreamento, e no capítulo seguinte, são apresentadas as continuidades desenvolvidas como contribuições desta tese.

3.1 Rastreamento online

Para realizar o rastreamento de objetos desconhecidos, a localização do alvo de interesse deve ser indicada no primeiro frame do vídeo. A partir deste momento, utilizando somente esta informação inicial, o sistema de rastreamento é treinado e executado em tempo real para localizar corretamente o objeto perseguido em cada um dos frames subsequentes. Todo o treinamento realizado neste rastreador ocorre em tempo real, sem a utilização de nenhum conhecimento prévio.

O sistema de aprendizado deste rastreador armazena as informações sobre os possíveis aspectos do objeto perseguido através de discriminadores, que são treina-

dos em tempo real, diretamente dos pixels da imagem. A cada frame do vídeo, os pixels pertencentes à localização retornada como resposta para o alvo são avaliados para determinar a necessidade de criar um novo discriminador ou realizar um novo treinamento com as informações dos pixels pertencentes à localização atual do alvo. Durante o rastreamento, o sistema mantém em memória, um conjunto de discriminadores que são utilizados para encontrar o alvo e são atualizados ou descartados sempre que necessário.

3.1.1 Binarização

Como apresentado na Subseção 2.1.1, o treinamento de um discriminador é realizado com um padrão de entrada convertido em uma sequência de bits. No caso do rastreamento de objetos, para realizar o treinamento de um padrão, o frame colorido é convertido em tons de cinza, com valores entre 0 e 255 (0 para pixels pretos e 255 para pixels brancos, com tons de cinza entre os dois valores). Posteriormente, a binarização é realizada com base no valor da luminância média dos pixels pertencentes à região retangular delimitadora do objeto (bounding box). Pixels com valores acima da luminância média recebem o valor 255, tornando-os pixels brancos, e os pixels com valores abaixo da luminância média recebem o valor 0, convertendo-os em pixels pretos. Com a imagem já binarizada, os pixels são convertidos em endereços para serem ativados nas memórias RAM de um discriminador.

3.1.2 Busca pelo alvo

Para cada um dos frames de um vídeo, a localização do objeto de interesse é passada para o frame seguinte, com o objetivo de realizar uma busca no entorno desta localização. Dessa maneira, dentro de uma janela de busca, as possíveis regiões retangulares do mesmo tamanho do alvo são convertidas em sequências de bits para serem classificadas em cada um dos discriminadores presentes no momento. A região com maior pontuação de classificação obtida dentro da janela de busca é escolhida para representar a localização do objeto a ser passada para o próximo frame, dando prosseguimento ao rastreamento. A pontuação retornada pelo melhor discriminador, obtida na região retornada como a possível localização para o alvo, é avaliada para verificar a necessidade de realizar alguma atualização na fila de discriminadores. A Figura 3.1 ilustra uma janela de busca em um determinado frame de índice $x+1$ obtida com base na resposta de localização retornada pelo frame x. Esse procedimento é realizado durante todo o rastreamento, em todos os frames do vídeo.

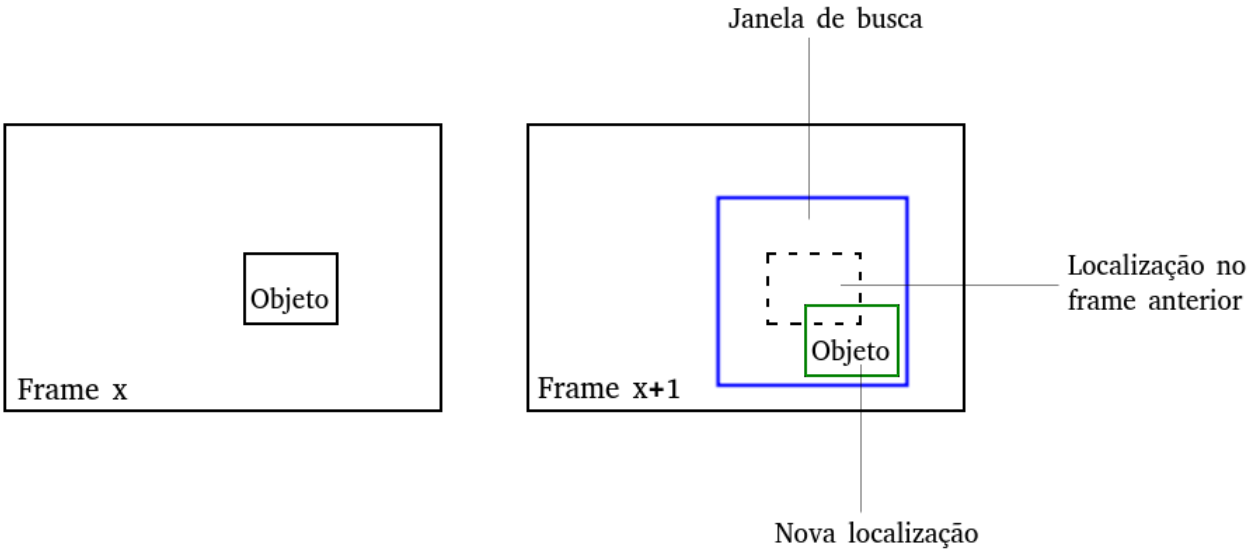

Figura 3.1: Janela de busca. No frame $x + 1$, a janela de busca é definida ao redor da localização retornada pelo rastreador no frame x. As possíveis novas localizações para o alvo são avaliadas em todos os discriminadores presentes na fila de discriminadores naquele momento.

3.2 Atualização da fila de discriminadores

A realização do rastreamento é possível devido ao armazenamento de múltiplos discriminadores representantes de um único objeto, que pode apresentar diferentes aspectos dentro de um mesmo conjunto de frames. Sendo assim, esta versão do rastreador armazena os discriminadores em uma fila, utilizando duas possibilidades de políticas de atualização de discriminadores.

3.2.1 Descarte de discriminadores há mais tempo sem utilização

A primeira abordagem de atualização dos discriminadores utiliza uma fila de armazenamento de discriminadores de tamanho fixo, onde, sempre que a fila estiver cheia e houver a necessidade de criar um novo discriminador, descarta-se o discriminador mais antigo para liberar espaço para o novo discriminador, pois assume-se que os discriminadores utilizados mais recentemente durante o rastreamento possuem maior chance de serem utilizados em um momento futuro próximo do que os discriminadores que já estão há mais tempo sem utilização. Sendo assim, os discriminadores mais atuais representam com mais eficiência, as aparências mais recentes apresentadas pelo objeto perseguido, e os discriminadores mais antigos representam os aspectos que estão há mais tempo sem aparecer, ficando guardados como se fossem memórias de longo prazo para serem utilizadas no momento oportuno em que

o objeto rastreado volte a apresentar uma aparência que já foi vista anteriormente.

Esse formato de gerenciamento da fila de discriminadores é iniciado no primeiro frame do vídeo, onde as coordenadas do objeto são passadas para o sistema, que realiza o treinamento de um discriminador com as informações do aspecto do objeto. No frame seguinte, realiza-se uma busca na vizinhança da localização retornada pelo frame anterior, buscando a localização com maior similaridade ao discriminador armazenado. Caso a nova localização do objeto seja classificada com uma pontuação abaixo de determinado *limiar de aceitação*, cria-se um novo discriminador para ser treinado com o aspecto atual, pois entende-se que o discriminador antigo está deixando de ser capaz de classificar corretamente os novos aspectos apresentados. Este novo discriminador é armazenado em uma fila de discriminadores, e nos frames subsequentes, a busca pelo objeto utiliza os dois discriminadores armazenados. Este processo se repete, atualizando-se a fila de discriminadores sempre que necessário, visando com que o rastreador se mantenha capaz de realizar o rastreamento corretamente em todos os frames do vídeo. Como a fila de discriminadores possui tamanho fixo, sempre que um novo discriminador é criado ou utilizado como responsável pela resposta de localização, ele é armazenado no início da fila, e dessa forma, naturalmente, os discriminadores há mais tempo sem utilização se encontram no final da fila, sendo removidos para liberar espaço sempre que necessário. A Figura 3.2 apresenta um exemplo de criação e atualização da fila de discriminadores para um conjunto de frames.

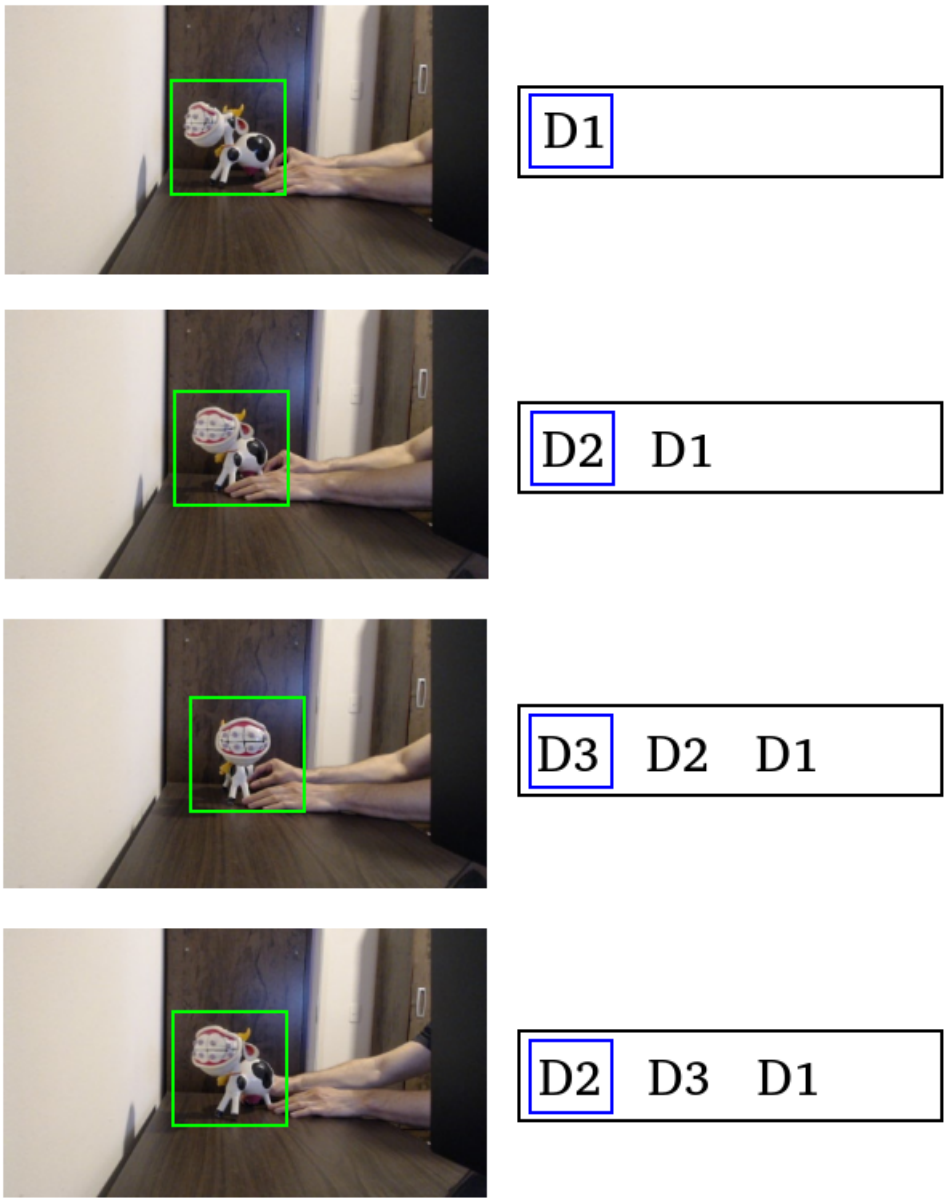

Figura 3.2: Primeira abordagem de atualização de discriminadores. No primeiro frame, D1 é responsável por localizar o alvo. No segunto frame, D1 já não retorna uma resposta adequada e então, D2 é criado e inserido no início da fila. No terceiro frame, D3 é criado pois o aspecto não é corretamente reconhecido nem por D1 e nem por D2. No quarto frame, D2 volta a retornar uma pontuação confiável, e é movido para o início da fila. Discriminadores mais recentemente utilizados sempre se encontram no início da fila.

3.2.2 Retreino de discriminadores

A abordagem apresentada anteriormente, realizava um único treinamento em cada discriminador armazenado na fila. Isto ocasionava a necessidade de criar novos discriminadores com muita frequência durante o rastreamento, pois um discriminador sendo utilizado em um determinado momento, rapidamente passava a retornar baixas pontuações de similaridade para os aspectos do objeto, mesmo com pequenas variações nos aspectos apresentados.

Visando a construção de um sistema de rastreamento mais robusto, utilizou-se uma segunda abordagem para a atualização de discriminadores, através de uma fila com armazenamento de tamanho fixo, com a possibilidade de realizar reforços de treinamento em um discriminador já existente, e realizando a criação de novos discriminadores somente enquanto houver espaço disponível. A estratégia de realizar um número predeterminado de retreinos em um mesmo discriminador, possibilitou que um único discriminador se tornasse mais resistente às mudanças nos aspectos dos objetos, diminuindo assim a frequência de criação de novos discriminadores.

O retreino de discriminadores ocorre de maneira similar ao algoritmo AutoWiSARD [47], onde são utilizados dois limiares, um *limiarAceitação* e um *limiarNovoDiscriminador*, que determinam as possíveis modificações a serem feitas nos discriminadores, como exemplificado na Figura 3.3. Sempre que a melhor resposta para a possível localização de um objeto rastreado se der por um discriminador em que a pontuação obtida seja maior que o *limiarAceitação*, não há a necessidade de criar nenhum novo discriminador e nem de realizar um retreino, pois assume-se que o discriminador está conseguindo representar o aspecto do objeto de maneira adequada. Caso o melhor discriminador retorne uma pontuação entre *limiarNovoDiscriminador* e *limiarAceitação*, então realiza-se um retreino neste discriminador, desde que o limite de retreinos por discriminador não tenha sido atingido, a fim de reforçar o conhecimento que descreve a forma atual com que o objeto se apresenta. E finalmente, caso a pontuação retornada pelo melhor discriminador retorne um resultado abaixo de *limiarNovoDiscriminador*, então cria-se um novo discriminador para representar o novo aspecto do objeto, caso exista espaço de armazenamento.

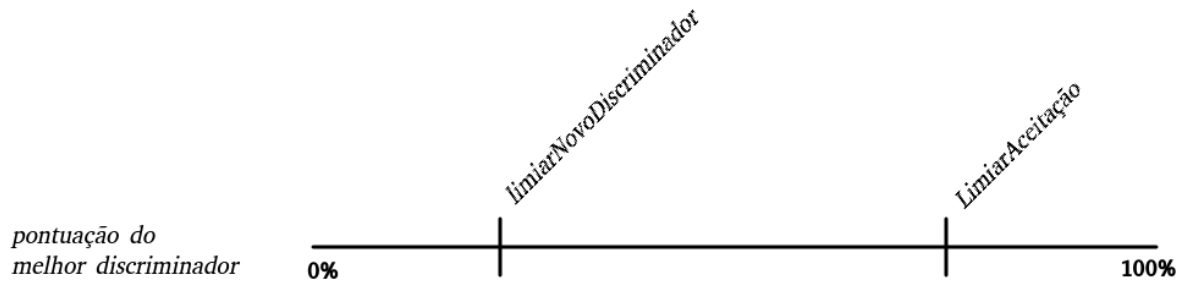

Figura 3.3: Limiares para atualização na fila de discriminadores. A pontuação obtida pelo melhor discriminador é utilizada para determinar uma das possibilidades: a criação de um novo discriminador, caso a similaridade esteja abaixo do *limiarNovoDiscriminador*; retreinar com o aspecto atual, caso a resposta esteja entre os dois limiares; ou não realizar nenhuma modificação, caso a resposta seja uma pontuação acima do *limiarAceitação*

A Figura 3.4 exemplifica os passos de atualização em uma fila de discriminadores de tamanho fixo com possibilidade de retreino para rastrear um determinado objeto. Diferentemente da abordagem anterior, não há mudanças na ordem dos discriminadores, pois após alcançar o número máximo de discriminadores, não ocorre mais a criação de nenhum discriminador novo, restando apenas a realização de possíveis retreinos para manter a capacidade de rastreamento do sistema. A partir do momento em que a ocupação máxima de armazenamento é alcançada juntamente com o máximo de treinos por discriminador, o rastreamento segue até o final com a mesma configuração na fila de discriminadores.

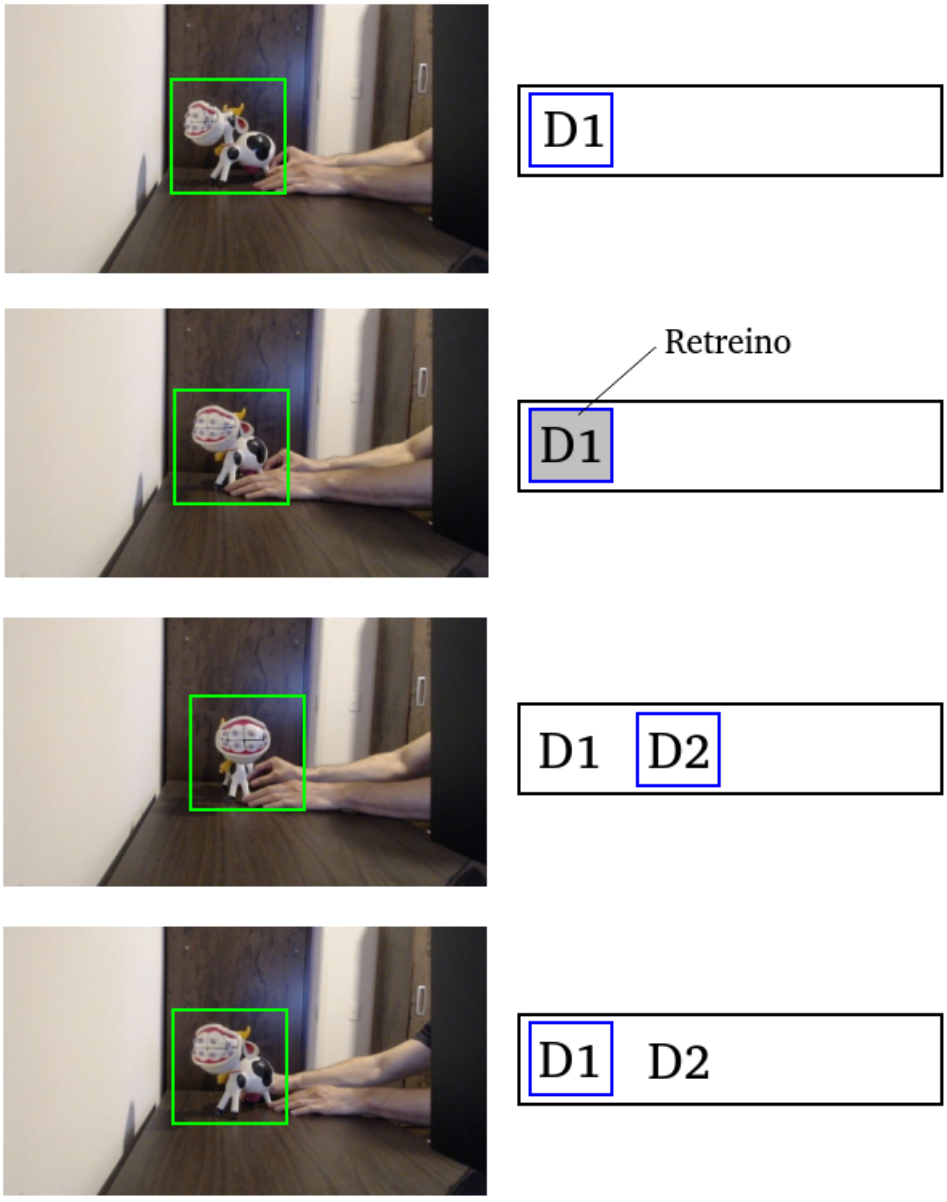

Figura 3.4: Segunda abordagem de atualização de discriminadores. No primeiro frame, D1 é responsável por localizar o alvo. No segunto frame, D1 retorna uma pontuação entre *limiarRetreino* e *limiarAceitação*, e sendo assim, recebe um reforço de treinamento. No terceiro frame, o discriminador D1 retorna uma pontuação abaixo do *limiarNovoDiscriminador*, de forma que, entende-se que está perdendo sua capacidade de reconhecer o objeto e cria-se então o discriminador D2. No quarto frame, o discriminador D1 volta a ser o responsável por localizar o objeto.

No presente trabalho, a abordagem de atualização da fila de discriminadores segue um misto entre as duas estratégias apresentadas, onde a fila de discriminadores possui tamanho fixo, e sempre que necessário, os discriminadores há mais tempo sem utilização são descartados, além de também ser possível realizar um determinado número de retreinos nos discriminadores. Neste trabalho, as entradas para a criação dos modelos são obtidas através do rastreamento dos objetos, e as evoluções realizadas neste rastreador são apresentadas no capítulo que se segue.

Capítulo 4

Evoluções no rastreador de objetos sem pesos

O presente trabalho apresenta uma série de evoluções que foram desenvolvidas em relação à versão original do rastreador de objetos sem pesos apresentada no capítulo anterior. Estas evoluções são apresentadas a seguir, e visam elaborar soluções para lidar com conhecidos problemas do rastreamento, como a perda da capacidade de rastreamento de objetos que se deslocam rapidamente para fora da região de busca ou até mesmo desaparecem do campo de observacão da câmera; problemas decorrentes de mudanças de escala com o afastamento ou aproximação do alvo em relação ao posicionamento da câmera; ou problemas de oclusão parcial que impedem o rastreador de identificar as localizações dos alvos corretamente. Dessa forma, este capítulo apresenta o desenvolvimento de um módulo detector para auxiliar o rastreamento, assim como a possibilidade de identificar mudanças de escala e oclusões parciais.

4.1 Detector

A versão inicial do rastreador realizava a busca dos objetos perseguidos em cada frame do vídeo, procurando em uma janela de busca ao redor da localização retornada no frame imediatamente anterior. Ao realizar esta busca local ao redor do alvo, poderia ocorrer a situação em que o objeto se desloca rapidamente para fora da região de busca, acarretando na perda do objeto pelo rastreador. Outra situação problemática se dava na situação de movimentação do objeto para fora do frame, fazendo com que o rastreador não conseguisse identificar novamente o objeto no seu retorno para dentro do frame, gerando a necessidade de desenvolver um detector de objetos para essa finalidade. Devido a essas situações, o presente trabalho adicionou a funcionalidade de detecção para auxiliar o sistema de rastreamento, realizando uma busca global, procurando o objeto nos frames em escala reduzida.

4.1.1 Busca em baixa resolução

O módulo rastreador realiza uma busca local nos frames do vídeo, no entorno da localização retornada pelo frame imediatamente anterior, mantendo a sua resolução original. Simultaneamente, o módulo detector realiza uma busca global, reduzindo os frames para baixas resoluções, sendo possível procurar em uma grande área do frame, e corrigir o rastreador, caso necessário. Dessa maneira, o detector efetua um papel de auxiliador do módulo rastreador. A Figura 4.1 ilustra os dois módulos executando simultaneamente no mesmo frame.

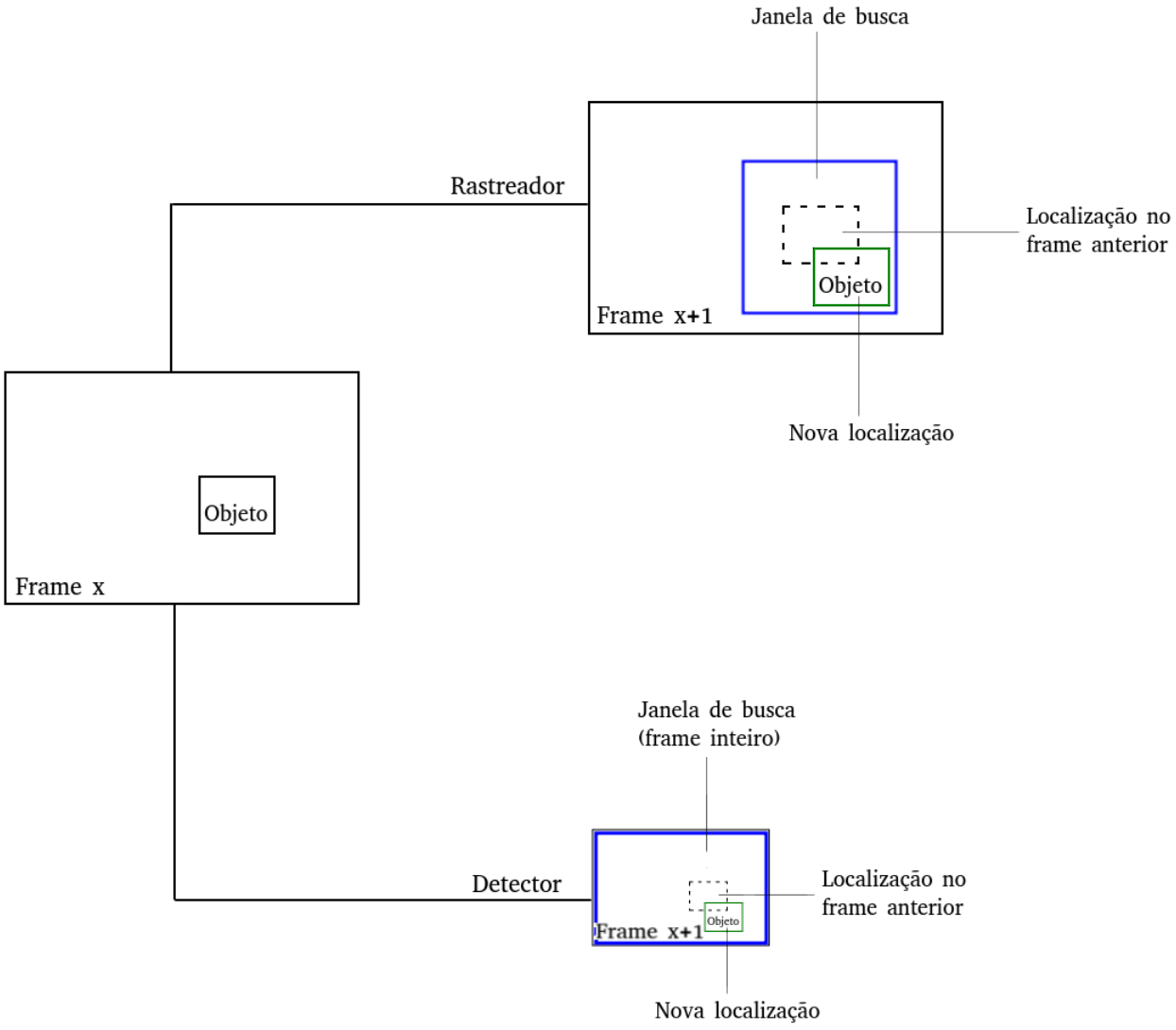

Figura 4.1: Rastreamento e Detecção. Cada módulo executa a busca nos mesmos frames simultaneamente, sendo uma busca local em alta resolução e uma busca global de baixa resolução.

4.1.2 Aspectos do detector

O módulo detector funciona de maneira semelhante ao rastreador, armazenando discriminadores para representar os aspectos apresentados pelo objeto rastreado. O detector possui uma fila de discriminadores de tamanho fixo, que é preenchida de acordo com a criação de discriminadores pelo módulo rastreador. Dessa forma, a cada novo aspecto visualizado pelo rastreador, um novo discriminador é treinado a partir da imagem reduzida e é adicionado na fila de discriminadores do detector. A Figura 4.2 ilustra essa formação de fila, onde para cada novo discriminador treinado e adicionado na fila do rastreador, o frame correspondente é redimensionado para uma escala reduzida, gerando uma aspecto reduzido que é utilizado para treinar um discriminador novo para ser adicionado à fila de discriminadores do detector. Esses discriminadores de tamanho reduzido são utilizados para realizar as possíveis classificações de localização buscando detectar o objeto na imagem reduzida.

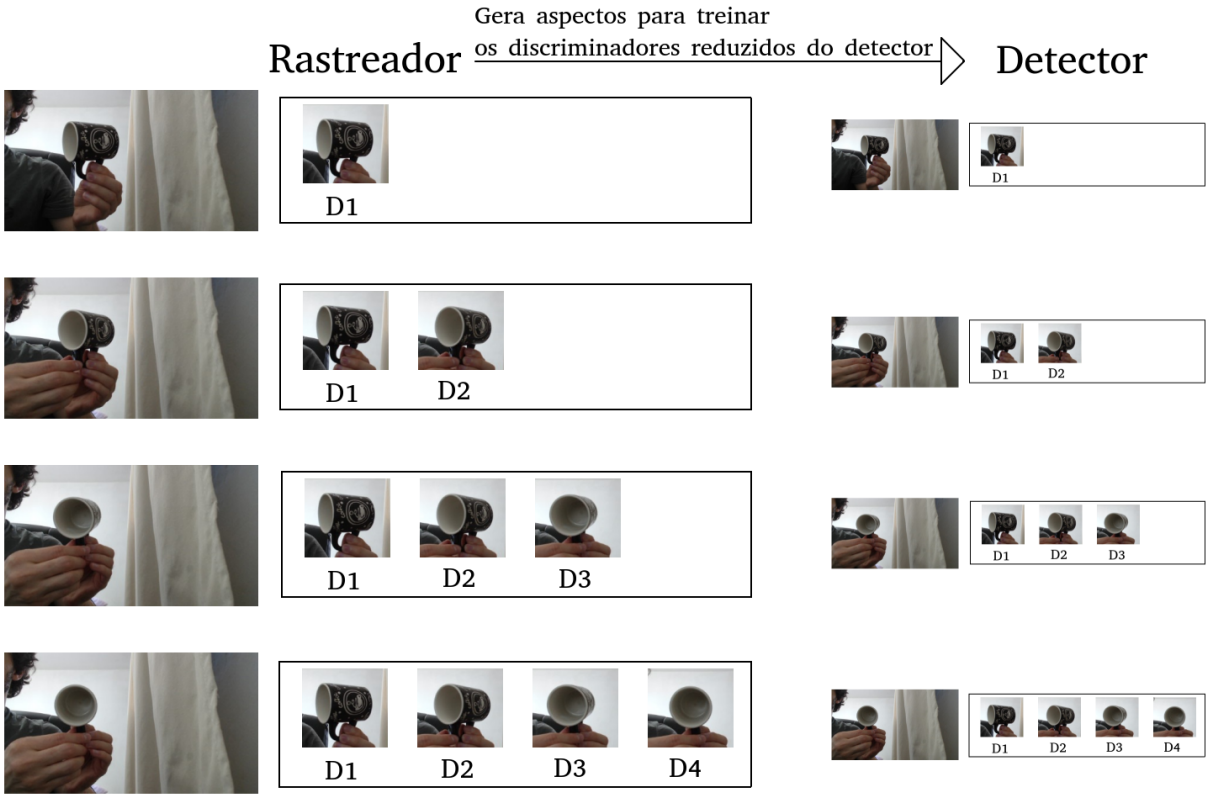

Figura 4.2: Fila de discriminadores do detector. Cada novo discriminador adicionado à fila do rastreador gera um discriminador para o detector, treinado a partir do frame reduzido.

A cada novo frame recebido, o detector procura o objeto em toda a imagem reduzida, e caso encontre, retorna a localização do objeto para o rastreador, a fim de verificar se existe a necessidade de corrigir a localização. A maneira utilizada para encontrar o objeto pelo detector segue o mesmo método empregado na busca realizada pelo rastreador, ou seja, para cada possível localização do alvo na imagem reduzida, os pixels são utilizados para formar a entrada a ser classificada em todos os discriminadores presentes na fila de discriminadores do detector. Então, a melhor pontuação obtida dentre todos os discriminadores, avaliados na imagem inteira, deve ser maior que um *limiarDetecção*, previamente determinado, a fim de considerar que uma detecção válida ocorreu. Caso tenha ocorrido uma detecção válida, a localização detectada é comparada com a localização retornada pelo rastreador para avaliar a necessidade de correção. Se os pontos centrais das regiões retangulares identificadas como alvo retornadas por ambos os módulos estejam localizados a uma distância maior que *limiarDistanciaCorreção*, então, o rastreador deve reiniciar o rastreamento nas coordenadas retornadas pelo detector. Abaixo, a imagem ilustra um rastreador perdendo o alvo perseguido e o detector recuperando a localização correta.

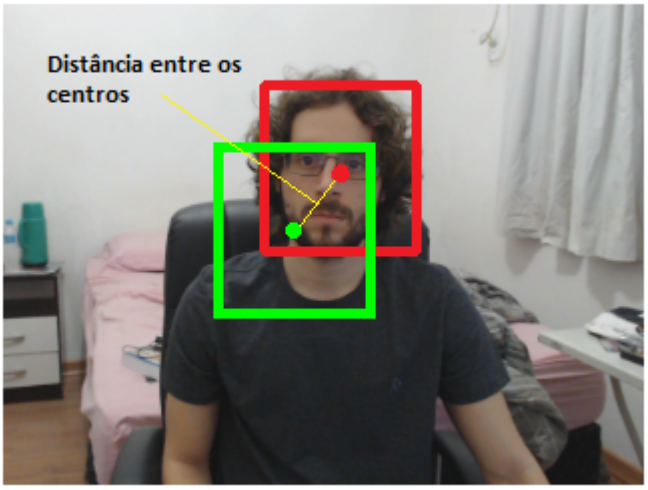

Figura 4.3: Correção do rastreador pelo detector. Neste exemplo, o rastreador se desviou da localização correta do objeto rastreado (marcação em verde) e o detector identificou o objeto a uma distância acima do limite estipulado por *limiarDistanciaCorreção* (marcação em vermelho). Então, a localização do rastreador sofre uma correção para os próximos frames.

4.1.3 Detecção em múltiplos tamanhos

O afastamento ou a aproximação do objeto rastreado em relação ao ponto de observação é um outro problema que ocorre com frequência e que é de grande importância para aplicações de rastreamento de objetos em vídeo. Ao ocorrer uma mudança na escala, automaticamente, o objeto observado passa a apresentar uma aparência distinta do objeto em tamanho original, considerando uma região delimitadora de mesmo tamanho. Dessa forma, este trabalho apresenta também a funcionalidade de detecção em múltiplas escalas, para ser adicionada ao sistema de rastreamento baseado no modelo WiSARD. O método de detecção utilizando discriminadores formados a partir de imagens reduzidas, possibilitou a criação de um detector para identificar múltiplas escalas de um mesmo objeto, realizando a busca nos frames reduzidos, a fim de manter uma boa performance para o sistema.

A detecção em múltiplas escalas é feita a partir do treinamento de discriminadores de tamanhos variados, responsáveis por armazenar o conhecimento representativo de um mesmo aspecto do objeto. Sendo assim, cada novo aspecto gera um grupo de discriminadores de diferentes tamanhos, que são armazenados na fila de discriminadores do detector. A criação de discriminadores de múltiplos tamanhos é feita redimensionando-se os frames originais para diversas escalas e treinando os discriminadores com os alvos redimensionados. A Figura 4.4 mostra um exemplo de frames utilizados para montar os discriminadores para os aspectos do objeto rastreado em diferentes tamanhos.

Figura 4.4: Redimensionamento de frames para treinamento do detector. Cada frame em tamanho original é redimensionado para treinar discriminadores para representar diferentes tamanhos de um mesmo aspecto. Neste exemplo, são utilizados 3 tamanhos para cada aspecto, e a configuração da fila de discriminadores resultante desse treinamento pode ser vista na Figura 4.5.

O treinamento de discriminadores a partir de diversos frames redimensionados resulta em um detector que possui uma fila de discriminadores de múltiplos tamanhos para cada aspecto visualizado do objeto. A quantidade de RAMs de um discriminador é determinada em função da quantidade de pixels presentes dentro do bounding box delimitador do objeto e do número de bits utilizados. Sendo **b** a quantidade de bits utilizada e **n** a quantidade de pixels dentro do bounding box do objeto, então, a quantidade de RAMs do discriminador é dada por **n**/**b**. No desenvolvimento deste detector, o número de bits é mantido fixo, então, discriminadores que representam um mesmo aspecto de um objeto em diferentes escalas possuem quantidades diferentes de RAMs. A Figura 4.5 mostra a configuração da formação de discriminadores obtida do treinamento dos frames redimensionados do exemplo anterior.

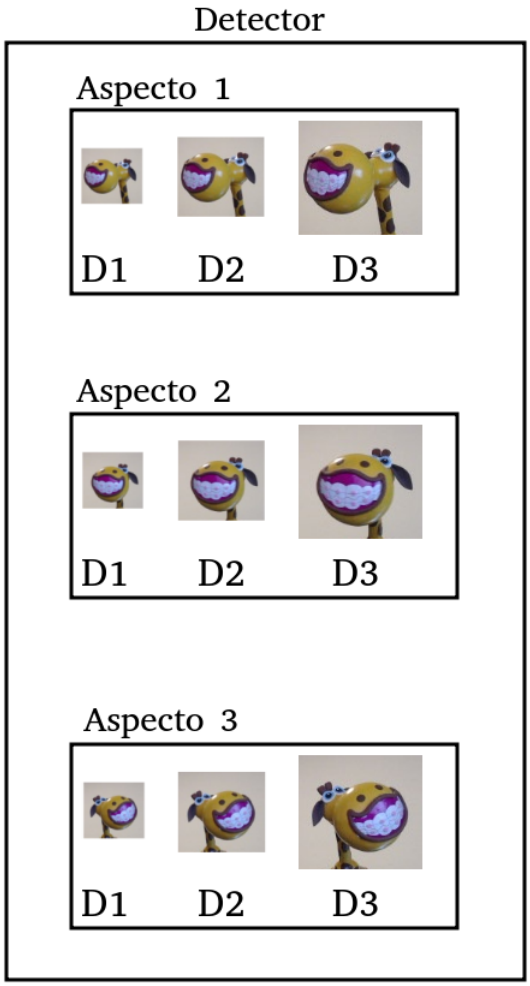

Figura 4.5: Fila de discriminadores do detector. Para cada aspecto observado, cria-se um novo discriminador correspondente a cada tamanho existente no detector. Cada frame do exemplo anterior é utilizado para treinar 3 discriminadores de tamanhos diferentes.

Nesta abordagem de detecção, em cada frame do vídeo, todos os discriminadores do detector são utilizados para procurar o objeto em uma imagem de baixa resolução, e aquele que retornar a maior porcentagem de ativação de RAMs, com o resultado acima de *limiarDetecção*, é utilizado para identificar o tamanho atual do objeto rastreado, juntamente com a sua localização. Como exemplo, os frames da imagem a seguir podem ser considerados, onde o rastreador atua buscando um determinado tamanho de objeto, e quando o detector identifica um afastamento do alvo observado, o rastreador é corrigido para procurar o objeto nos próximos frames, considerando a nova escala identificada.

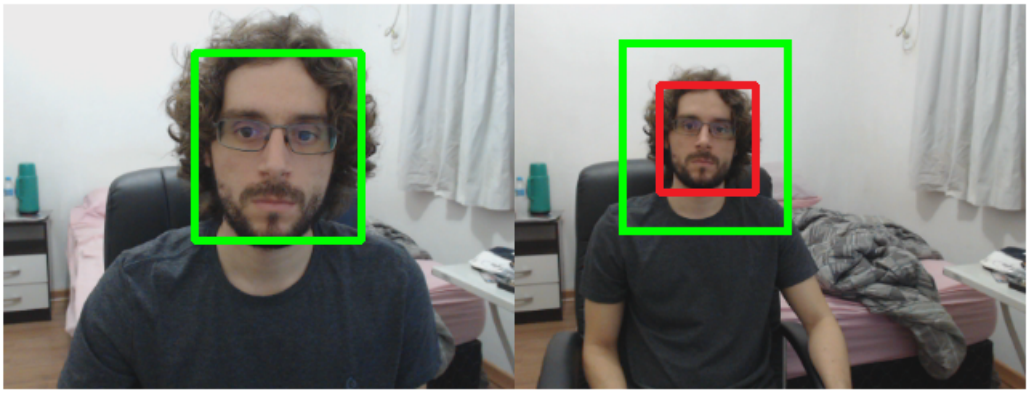

Figura 4.6: Correção de escala. No frame da esquerda, o rosto rastreado está marcado com a resposta retornada pelo rastreador. No frame da direita, o detector identificou que o rosto mudou de escala (marcação em vermelho), e assim, nos próximos frames, o rastreamento é reiniciado para buscar o objeto na nova escala identificada.

O algoritmo de detecção é apresentado a seguir, onde a entrada é um conjunto de discriminadores, um de cada tamanho, todos representando aspectos de um mesmo objeto e a busca é efetuada na imagem em baixa resolução, porém, mantendo-se o tamanho dos discriminadores, onde cada discriminador representa o alvo em determinada escala. Caso algum deles retorne valor de ativação acima do limiar, assume-se que o objeto foi detectado na escala correspondente, e assim, retorna-se a localização e a escala para a imagem original.

Algoritmo 2 Detecção de um objeto

Entrada: Fila de Discriminadores do Detector e frame atual

$frameAtualReduzido \leftarrow reduzFrameAtual(frameAtual)$

para cada $discriminador \in filaDiscriminadoresDetector$ **faça**

$buscaAlvoNaImagemReduzida(discriminador,\ frameAtualReduzido)$

se $ativacaoDiscriminador > limiarDeteccao$ **então**

Objeto Detectado na escala do *discriminador*.

fim se

fim para

4.2 Integração rastreador-detector

As operações de rastreamento e detecção ocorrem simultaneamente em cada frame do vídeo, com o objetivo de tornar o rastreamento mais robusto, possibilitando a recuperação em casos de perda da localização do alvo de interesse. Sendo assim, o sistema desenvolvido trabalha com uma thread para realizar o rastreamento e outra para realizar a detecção, ambas executando as suas tarefas para cada um dos frames recebidos como entrada. A imagem a seguir ilustra a integração do rastreador com o detector, e o algoritmo implementado é apresentado na sequência.

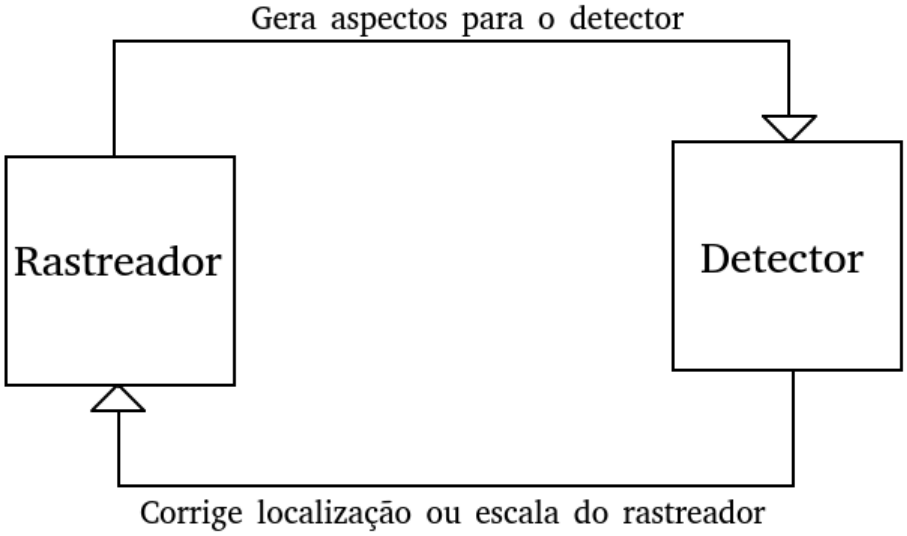

Figura 4.7: Integração Rastreador-Detector. O rastreador realiza a busca localmente criando aspectos que são adicionados em diferentes escalas na fila de discriminadores do detector. O detector identifica a localização e a escala do alvo, corrigindo o tracker quando necessário.

Algoritmo 3 Integração Rastreador-Detector

Entrada: Coordenadas iniciais do objeto alvo

Entrada: Sequência de frames contendo o objeto alvo

```
para cada frame de entrada do vídeo faça
    Executa thread de rastreamento
    Executa thread de detecção
    se Novo discriminador adicionado ao rastreador então
        se Há espaço de armazenamento disponível no detector então
            Cria discriminadores redimensionados para o detector
        fim se
    fim se
    se ObjetoDetectado então
        se EscalaDetector ≠ EscalaRastreador então
            Corrige escala do rastreador
        fim se
        se CoordenadasDetector ≠ CoordenadasRastreador então
            Corrige localização do rastreador
        fim se
    fim se
fim para
```

4.3 Identificação de oclusão parcial

A oclusão do objeto rastreado é um dos grandes dificultadores para realizar o rastreamento, pois pode fazer com que informações erradas sejam absorvidas pelos sistemas de aprendizado, ocasinando a perda de capacidade de encontrar a localização correta do alvo original. Dessa forma, este trabalho apresenta uma abordagem de utilização de subdiscriminadores para identificar possíveis situações de oclusão. Assim como o rastreador e o detector descritos anteriormente, a identificação de uma oclusão parcial também utiliza múltiplos discriminadores, porém, dividindo um mesmo alvo em partes que geram discriminadores diferentes e que estão relacionados entre si. Desta forma, para cada discriminador treinado no rastreador, associa-se um grupo de subdiscriminadores, responsáveis por identificar partes do alvo, assim como ilustrado na figura a seguir.

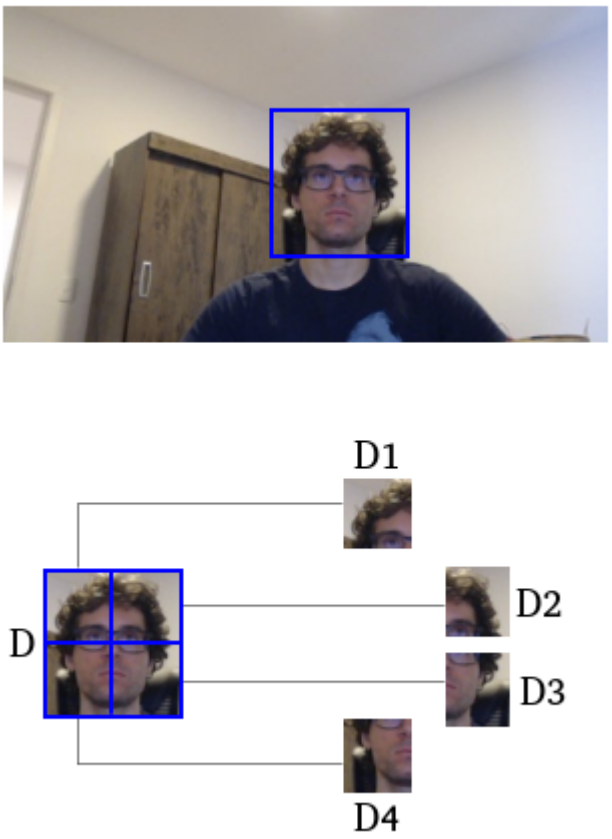

Figura 4.8: Formação de subdiscriminadores. O aspecto apresentado gera o treinamento do discriminador D, e de seus subdiscriminadores, D1, D2, D3 e D4, formados por partes do aspecto.

Como exemplo, considere um objeto de interesse que seja dividido em quadrantes e cada quadrante gere um discriminador diferente. Dessa forma, tem-se quatro discriminadores que representam quatro partes diferentes do objeto. Caso o objeto não esteja sofrendo nenhuma oclusão durante o rastreamento, espera-se que os quatro discriminadores retornem uma quantidade alta de RAMs ativadas, porém, supondo que um outro objeto esteja entrando na frente do alvo, espera-se que os discriminadores pertencentes ao lado que está sendo ocluso retornem pontuações baixas, enquanto os discriminadores correspondentes às partes não escondidas devem continuar retornando pontuações altas. Desta maneira, é possível detectar a partir de qual localização relativa ao alvo, uma oclusão ocorre.

A identificação da ocorrência de uma oclusão é importante para evitar que novos discriminadores sejam criados a partir de partes do frame não correspondentes ao objeto de interesse, evitando que o sistema incorpore informações errôneas de aprendizado e acabe se perdendo. O algoritmo 4 mostra como é realizada a identificação de oclusão parcial baseada no uso de subdiscriminadores, onde o alvo é dividido em quadrantes, e um discriminador é treinado para cada parte do objeto. Se em deter-

minado momento, alguns dos subdiscriminadores retornam ativação abaixo do *limiar de oclusão* e os outros retornam ativação acima do *limiar de aceitação*, considera-se que uma oclusão parcial está acontecendo, podendo evoluir para uma oclusão total. Neste caso, evita-se que ocorram novos treinamentos de discriminadores para o rastreador, a fim de que padrões errados não sejam aprendidos, evitando assim que o rastreador acabe se perdendo.

Algoritmo 4 Detecção de Oclusão

1: treinaDiscriminadorParaCadaQuadrante($frameAtual, localizaoObjeto$)
2: **se** ($\exists(QuadranteComAtivacao < limiarOclusao)$ & $\exists(QuadranteComAtivacao > limiarAceitacao)$) **então**
3: Oclusão parcial detectada
4: **fim se**

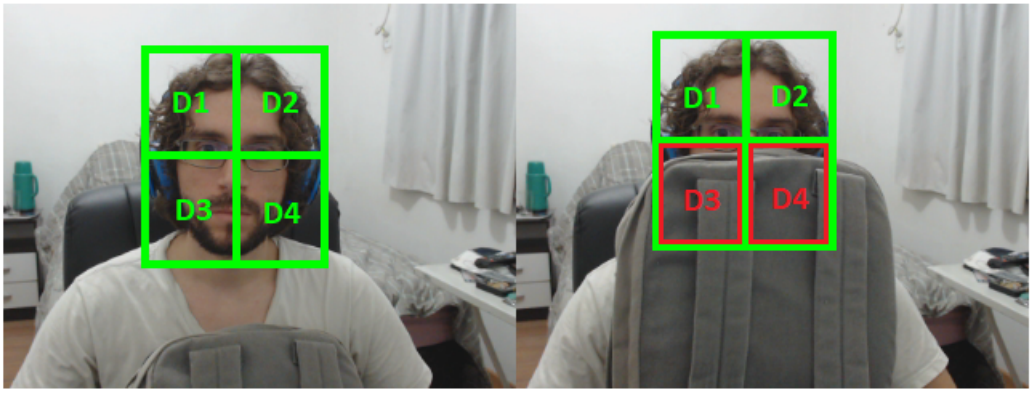

Figura 4.9: Detecção de oclusão. O alvo é dividido em partes, cada uma associada a um subdiscriminador. No frame da esquerda, todos os subdiscriminadores retornam uma pontuação alta, indicando que todas as partes do objeto foram identificadas. No frame da direita, os subdiscriminadores **D1** e **D2** retornam pontuações altas e os subdiscriminadores **D3** e **D4** retornam pontuações baixas. Então, o sistema assume que está ocorrendo uma oclusão nesta parte da imagem.

Capítulo 5

Modelador de objetos

O método utilizado para a criação dos modelos dos objetos a partir das imagens mentais dos discriminadores WiSARD é apresentado neste capítulo. O rastreador de objetos apresentado no capítulo anterior consegue obter as localizações do objeto observado nos frames de um vídeo, porém, não é capaz de fornecer nenhum tipo de entendimento sobre as estruturas visuais dos aspectos, e sendo assim, o objetivo desta tese é a criação em tempo real de modelos que representem objetos rastreados em vídeo, fornecendo o entendimento sobre os aspectos e as possibilidades de transições entre cada aspecto aprendido. Para a realização da criação dos modelos, o rastreador de objetos apresentado no capítulo anterior foi utilizado para obter as localizações do objeto de interesse e passá-las como entradas para o algoritmo ClusWiSARD, responsável por realizar o aprendizado dos aspectos em conjunto com as possíveis transições. Portanto, a nova funcionalidade de criação de modelos foi incorporada ao rastreador, mantendo a característica de realizar todo o processamento em tempo real, sem a necessidade de treinamento prévio.

5.1 Algoritmo de modelagem

A modelagem dos objetos ocorre a partir das localizações retornadas pelo rastreador em cada um dos frames de vídeo. Essas localizações são passadas para a ClusWiSARD que realiza o aprendizado dos aspectos observados, determinando se um novo discriminador deve ser criado para receber o novo aspecto, ou se este deve ser treinado em um discriminador já existente, caso seja similar a um aspecto já visto anteriormente durante o rastreamento. Desta forma, é possível criar um mapeamento das transições entre os aspectos aprendidos pela ClusWiSARD, considerando a ordem de criação de novos aspectos ou de visualização de aspectos já aprendidos.

No algoritmo original da ClusWiSARD, cada novo padrão de entrada deve ser classificado por todos os discriminadores de classe correspondente a fim de avaliar quais discriminadores devem absorver o novo padrão. Caso a pontuação retornada

por um discriminador de classe correspondente ao padrão de entrada seja maior ou igual a $\boldsymbol{min(1, s + size(d)/\gamma)}$, sendo $\boldsymbol{s}$ a pontuação de similaridade mínima, $\boldsymbol{size(d)}$ a quantidade de padrões absorvidos pelo discriminador, e $\boldsymbol{\gamma}$ o intervalo de crescimento, assim como apresentado na Seção 2.3, então, este discriminador é treinado com o novo padrão de entrada. Caso o padrão de entrada não tenha sido absorvido por nenhum discriminador, então, um novo é criado para recebê-lo. Entretando, existe a situação em que existem clusters similares ao novo padrão de entrada, porém, todos esses clusters já alcançaram o seu limite de treinamento, sendo necessário criar um novo cluster para absorver esse padrão.

Essa situação acarreta na existência de mais de um cluster para representar as características de padrões muito similares, causando problemas para a criação de modelos, pois como os aspectos são aprendidos a partir das entradas obtidas pelo rastreamento em tempo real, pode acontecer de um objeto rastreado se manter estacionário por um período de tempo, atingindo a saturação de um discriminador representante desse aspecto, forçando com que novos discriminadores sejam criados para receber um aspecto já conhecido. Isto é problemático pois torna os modelos de transições mais complexos de serem processados e entendidos, gerando estados com informações repetidas e criando transições desnecessárias.

A solução encontrada envolve a criação de um *limiar de aspecto visto* para evitar a criação de discriminadores desnecessariamente, gerando estados muito similares. Neste caso, modificando o algoritmo original da ClusWiSARD, caso nenhum discriminador tenha absorvido o novo padrão de entrada, então, verifica-se se este padrão já foi visto anteriormente através da melhor pontuação obtida por sua classificação em cada um dos discriminadores. Se esta pontuação for um valor acima do *limiar de aspecto visto*, então, considera-se que este aspecto já foi visto e nenhum novo discriminador é criado, descartando-se este novo aspecto. Outra possibilidade para resolver este problema seria aumentar a capacidade de treinamento para cada um dos discriminadores, porém, ao receber um grande número de aspectos, um mesmo discriminador passa a não fornecer características tão boas para representar um único aspecto, pois acaba generalizando em demasia e retornando imagens mentais distorcidas e não adequadas para utilização no modelo de estados de aspectos.

Além do aprendizado dos aspectos através dos clusters formados, o algoritmo de modelagem envolve a criação de transições entre aspectos. Isto é feito através da ordenação de aprendizagem e visualização dos padrões, onde para cada novo aspecto aprendido, é criada uma transição para o último aspecto visto pelo sistema, e também é criada a transição no sentido oposto, entre o último aspecto visualizado e o novo aspecto aprendido. Além disso, caso nenhum discriminador tenha aprendido o novo padrão de entrada e este já tenha sido visualizado anteriormente no rastreamento, então, cria-se uma transição entre o aspecto mais similar ao novo padrão de

entrada e o último aspecto visto pelo sistema, e também adiciona-se a transição no sentido oposto. Os algoritmos a seguir descrevem o funcionamento do modelador de objetos, que recebe as entradas do rastreador e executa o algoritmo ClusWiSARD para aprender os aspectos e adicionar possíveis transições aos modelos de estados.

Algoritmo 5 Rastreamento e obtenção de entradas para o modelador de objetos baseado na ClusWiSARD

Entrada: Coordenadas iniciais do objeto alvo

Entrada: Sequência de frames contendo o objeto alvo

para cada frame de vídeo **faça**

$localizacaoAtualObjeto \leftarrow$ RASTREAMENTO($frameAtual$)

$aspectoDeEntrada \leftarrow$ MAPEAMENTO($localizacaoAtualObjeto, frameAtual$)

MODELADORCLUSWISARD($aspectoDeEntrada$)

fim para

Algoritmo 6 Modelador de Objetos ClusWiSARD

Entrada: $aspectoDeEntrada$ = aspecto de entrada mapeado

Entrada: s = similaridade mínima

Entrada: γ = intervalo de crescimento

Entrada: δ = limiar de objeto visto anteriormente

```
Determine o discriminador com maior valor de ativação para o padrão de entrada
para cada discriminador d presente no modelo do objeto faça
    se score(d, aspectoDeEntrada) ≥ min(1, s + size(d)/γ) então
        Discriminador d aprende inputAspect
        size(d) ← size(d) + 1
    fim se
fim para
se Nenhum discriminador aprendeu aspectoDeEntrada então
    se ScoreDiscriminadorMaisSemelhante ≤ δ então
        Um novo discriminador d' é criado
        d' aprende aspectoDeEntrada
        size(d') ← 1
        d' é adicionado ao modelo do objeto
        ADICIONATRANSICAO(ultimoAspectoVisualizado, aspectoDeEntrada)
        ADICIONATRANSICAO(aspectoDeEntrada, ultimoAspectoVisualizado)
    senão
        ADICIONATRANSICAO(ultimoAspectoVisualizado, aspectoMaisSimilar)
        ADICIONATRANSICAO(aspectoMaisSimilar, ultimoAspectoVisualizado)
    fim se
fim se
```

5.2 Modelos de imagens mentais

O algoritmo de modelagem agrupa os aspectos rastreados em clusters representados por discriminadores, determinados pelo algoritmo ClusWiSARD. Como cada cluster de aspecto é formado por um discriminador, é possível visualizar o conhecimento armazenado em cada discriminador através da obtenção de imagens mentais, como descrito na Seção 2.4. A junção das imagens mentais representantes dos aspectos, com as transições aprendidas, determinadas pelo algoritmo de criação de modelos, resulta em um grafo de estados de aspectos, onde é possível identificar as relações entre os aspectos, assim como possíveis caminhos entre aspectos distintos. A Figura 5.1 ilustra um grafo de estados obtido a partir de um rosto rastreado, onde é

possível entender a movimentação que foi realizada durante o rastreamento para a obtenção deste modelo. Em seguida, na Figura 5.2 pode ser observado um grafo de estados obtidos a partir do rastreamento de uma xícara. Como pode ser observado, as imagens mentais obtidas diretamente dos discriminadores determinados pelo agrupamento de aspectos através da ClusWiSARD geram representações que fornecem relações entre os aspectos de um mesmo objeto, adicionando conhecimento que pode ser utilizado pelo rastreador, que originalmente de maneira resumida, acompanhava as mudanças nos aspectos, através das mudanças nos pixels das imagens, porém, sem conseguir obter uma compreensão mais profunda sobre as mudanças que acontecem nos cenários observados.

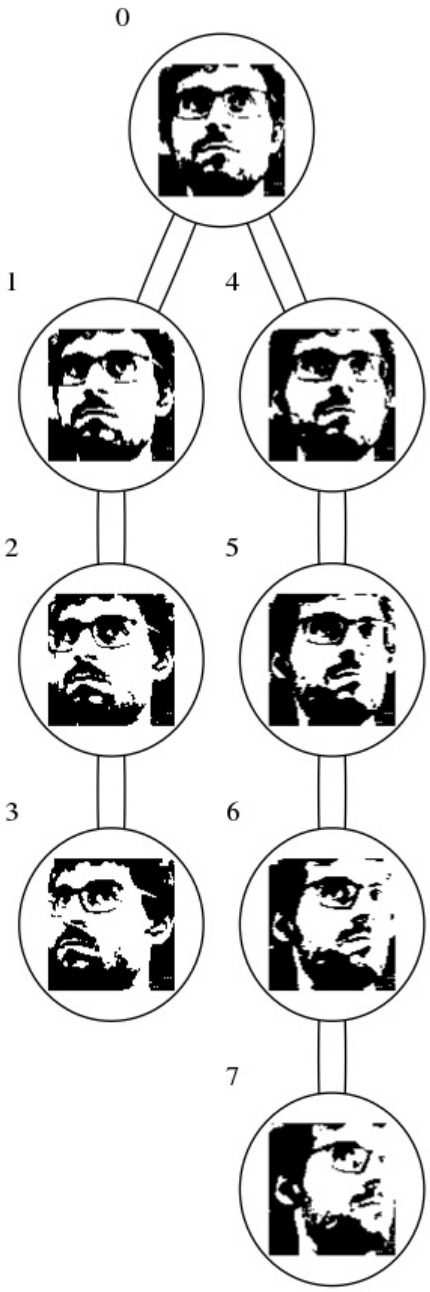

Figura 5.1: Modelo de um rosto. Modelo gerado em tempo real a partir do rastreamento de um rosto. Neste exemplo, o aspecto 0 é o primeiro aspecto aprendido. Em seguida, o rosto se movimenta para a esquerda, e o modelo aprende os aspectos 1, 2 e 3, adicionando as transições (0, 1), (1, 0), (1, 2), (2, 1), (2, 3) e (3, 2) às transições do modelo. Neste momento, o rosto se encontra no estado 3 e faz uma movimentação para a direita, realizando o caminho de volta no grafo 3->2->1->0, retornando para o estado inicial. Nesta volta, nenhum novo aspecto é adicionado ao modelo. Em seguida, o rosto continua a sua movimentação para a direita, criando as transições (0, 4), (4, 0), (4, 5), (5, 4), (5, 6), (6, 5), (6, 7) e (7, 6) juntamente com os novos aspectos.

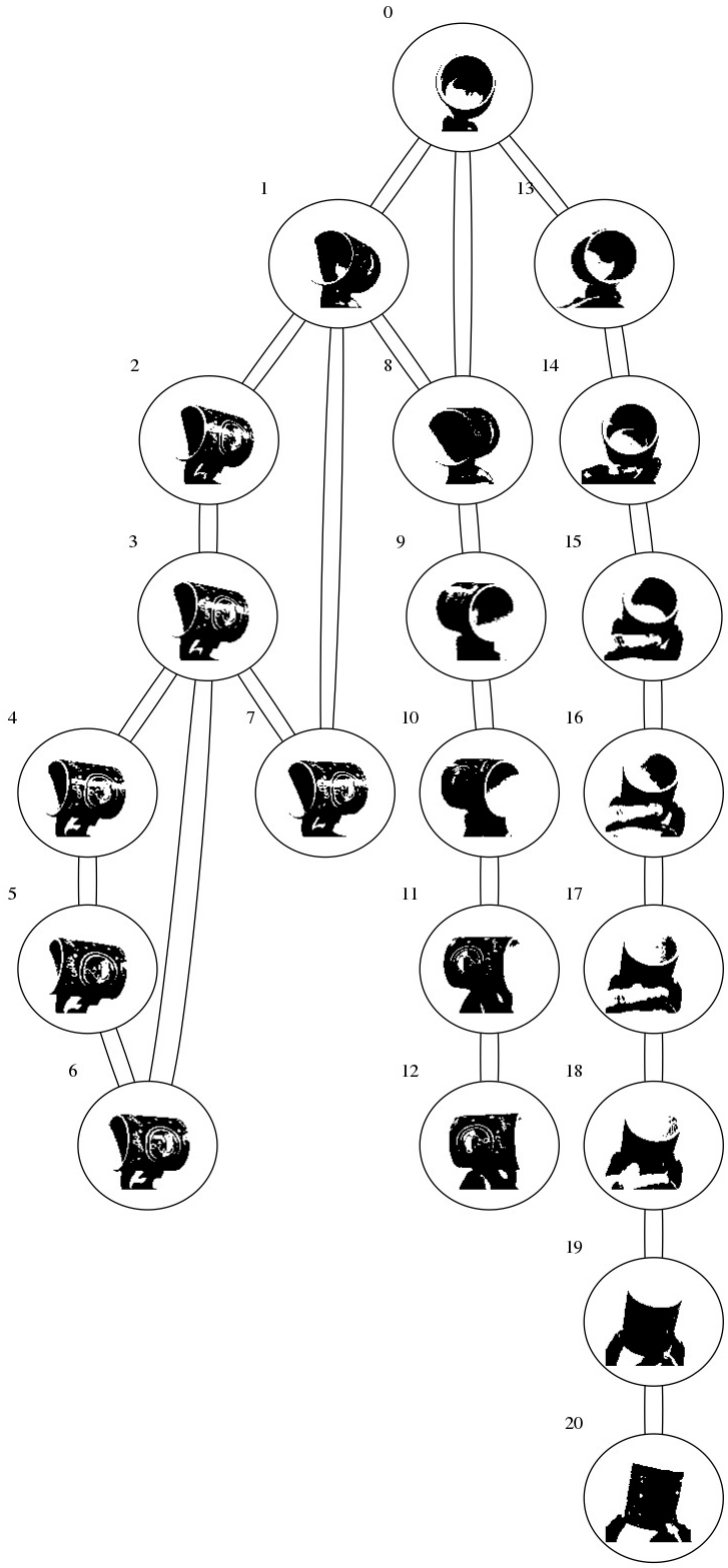

Figura 5.2: Modelo de uma xícara. Grafo de estados gerados a partir do rastreamento de uma xícara movimentada por uma mão.

5.3 Integração do sistema

O capítulo anterior apresentou o rastreador em funcionamento com o auxílio do detector. Esses módulos são executados em todos os frames do vídeo e se complementam, de maneira que para um determinado frame, o processamento de ambos deve ser finalizado antes de passar para o rastreamento no frame seguinte. Porém, a funcionalidade de modelagem de objetos baseada no algoritmo ClusWiSARD, pode ser executada em background, sem a necessidade de que o rastreamento espere a finalização do processamento executado pelo módulo modelador antes de seguir para o próximo frame.

Dessa forma, em cada frame de vídeo, o aspecto identificado pelo rastreador é colocado em em uma fila no buffer de processamento acessado pelo módulo modelador, que avalia cada um desses aspectos através do algoritmo ClusWiSARD, construindo os modelos de imagens mentais. Portanto, o sistema por completo possui três threads de execução principais, uma para o rastreador e uma para o detector, onde há a necessidade de esperar a finalização do processamento de um frame para dar prosseguimento ao frame seguinte, e uma thread de execução independente que é responsável pela modelagem do objeto, realizada em background. A imagem a seguir ilustra a integração do sistema.

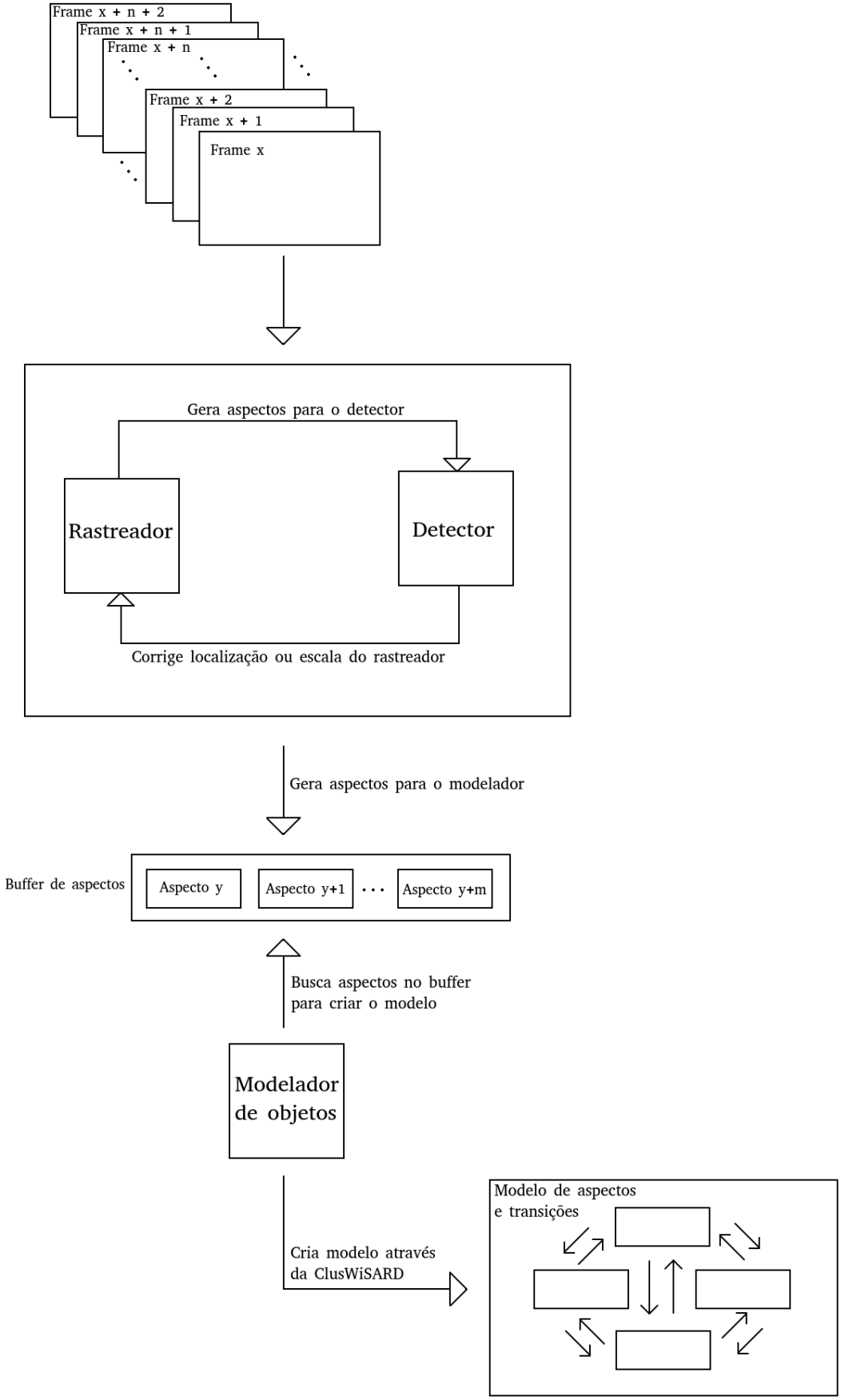

Figura 5.3: Integração completa do sistema. Rastreador e detector localizam o objeto e geram aspectos para o modelador que executa a criação do modelo em background através do algoritmo ClusWiSARD.

Capítulo 6

Experimentos e Resultados

Este capítulo apresenta os experimentos realizados para avaliação dos modelos mentais apresentados no capítulo anterior, com o objetivo de verificar se as representações de aspectos através de imagens mentais são adequadas e se realmente é possível obter informações relevantes sobre as relações entre os aspectos dadas pelas possíveis transições entre estados.

6.1 Datasets

Esta seção aborda os dois datasets que foram utilizados para realização de experimentos.

6.1.1 Object Modeling Through Weightless Tracking Dataset

Para realizar as avaliações da qualidade dos modelos, o dataset Object Modeling Through Weightless Tracking Dataset [57], disponível em https://doi.org/10.6084/m9.figshare.24034317.v1, foi utilizado. Este dataset foi uma das contribuições desta tese e possui uma série de frames de vídeos gravados através de uma webcam, cada vídeo fornecendo destaque para um objeto a ser rastreado e modelado em tempo real. Como o objetivo principal do sistema proposto é a criação dos modelos de objetos rastreados, outros datasets com vídeos possuindo grandes desafios para o rastreamento não se mostraram adequados para criar os modelos, visto que movimentações muito bruscas, mudanças repentinas nas aparências, oclusões dos objetos e muitas mudanças na escala não fornecem uma visualização adequada dos aspectos, impedindo o aprendizado de uma interpretação adequada sobre as estruturas visuais dos objetos e das relações entre os aspectos observados. Desta forma, fez-se necessário criar um dataset específico onde os objetos são apresentados para a câmera se movimentando de forma que transitem entre as suas possíveis aparências, tornando viável o aprendizado dos modelos.

6.1.2 OTB100

O outro dataset utilizado neste trabalho foi o OTB100 [58], um dataset amplamente utilizado para avaliação de rastreadores de objetos em vídeo. Este dataset consiste em um conjunto de 100 vídeos possuindo os mais diversos desafios encontrados em problemas de rastreamento de objetos, como mudanças na aparência do alvo, mudanças de escala, oclusão, variação de iluminação, problemas com imagens desfocadas e objetos que se movimentam com grande velocidade. Como este dataset possui uma ampla variedade de dificuldades a serem solucionadas para rastrear os objetos corretamente em vídeo, este foi utilizado para avaliar apenas a tarefa de rastreamento, onde o objetivo é identificar corretamente os alvos de interesse em cada um dos frames dos vídeos. Este dataset possui ainda um arquivo ground truth para cada um dos vídeos com anotações informando as localizações corretas do alvo em cada frame, através de um bounding box. Desta forma é possível avaliar a acurácia do rastreamento considerando uma ampla variedade de cenários.

6.2 Métricas de avaliação

O objetivo principal deste trabalho se dá na construção de modelos que representem adequadamente os objetos rastreados em vídeo. Desta forma, alguns dos experimentos que seguem nas próximas seções apresentam resultados qualitativos, onde pode-se observar resultados visuais obtidos através dos modelos. Além das avaliações qualitativas, as duas métricas que foram utilizadas para avaliar o sistema apresentado foram a similaridade entre pixels, e a métrica Intersection over Union, descritas a seguir.

6.2.1 Similaridade entre pixels

Além dos resultados visuais que serão apresentados nas seções seguintes, a avaliação da qualidade das imagens geradas pelos modelos foi avaliada através da similaridade entre pixels, onde para cada frame dos vídeos analisados, calcula-se a similaridade pixel a pixel, comparando a imagem binarizada correspondente ao objeto no frame original com a imagem mental correspondente identificada pelo sistema, obtida diretamente do modelo mental criado. A similaridade entre pixels é calculada frame a frame, comparando-se o aspecto do objeto no frame original com o correspondente aspecto identificado pelo modelo mental. Sendo assim, a taxa de acerto de pixels em um frame é obtida de acordo com as fórmulas a seguir, onde *ObjetoBinarizado(x, y)* e *ImagemMental(x, y)* correspondem aos valores dos pixels obtidos na coordenada *(x, y)* do frame avaliado.

$$TaxaAcertoFrame = 1 - (PixelsErradosFrame/totalPixelsFrame)$$

$$PixelsErradosFrame = \sum_{\forall x, \forall y} (ObjetoBinarizado(x, y) - ImagemMental(x, y))^2$$

6.2.2 Intersection over Union

A métrica de avaliação Intersection over Union (IoU) [59] foi utilizada para avaliar a acurácia do rastreamento, considerando as localizações retornadas pelo rastreador e os gabaritos das localizações dos objetos em cada um dos frames dos vídeos avaliados. Para um determinado frame, a métrica IoU é calculada determinando-se a área de interseção entre o bounding box previsto e o bounding box que representa a resposta correta de localização do alvo. Para cada frame de um vídeo, considera-se então um acerto de localização de rastreamento quando o valor de IoU for maior que determinado limiar. A Figura 6.1 ilustra o cálculo do valor de IoU para um bounding box previsto em relação a um bounding box gabarito.

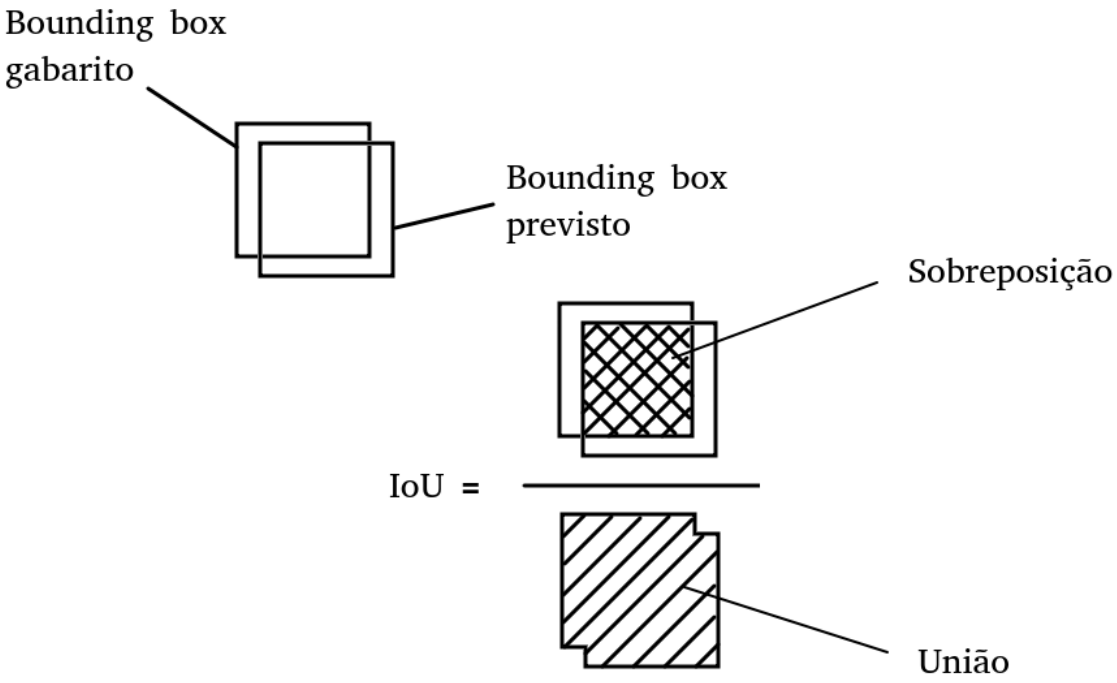

Figura 6.1: Intersection over Union. Métrica para cálculo de acurácia nos rastreadores de objetos, onde para cada frame, mede-se a sobreposição entre previsão e gabarito, dividindo o resultado pela união entre previsão e gabarito.

6.3 Avaliação dos aspectos aprendidos

Esta seção visa avaliar a qualidade das representações dos aspectos através dos modelos de imagens mentais. Para isso, foi desenvolvida uma aplicação que realiza o rastreamento baseado em um modelo de objeto concluído, sem executar novas atualizações durante o rastreamento. Esta aplicação utilizou um determinado conjunto de frames para realizar a modelagem do objeto rastreado, e em seguida, em um outro conjunto de frames, utilizou o modelo criado para detectar e rastrear o objeto modelado.

6.3.1 Rastreamento através de modelo

Para realizar o rastreamento através de um modelo pronto, primeiramente deve-se treinar o modelo de aspectos e transições a partir do rastreamento em um conjunto inicial de frames. A partir do momento em que o modelo está pronto, este é utilizado para buscar o objeto alvo em um conjunto de frames nunca vistos anteriormente, sem utilização de mais nenhum tipo de treinamento durante a execução.

As informações obtidas através do modelo pronto são as únicas a serem utilizadas para realizar todo o rastreamento no novo conjunto de frames, sem nem mesmo utilizar as informações de localização inicial. Para isso, foi necessário utilizar uma detecção através do modelo. Então, antes de realizar o rastreamento, cria-se uma instância de detector formada a partir dos discriminadores do modelo. Esse detector busca o objeto nos primeiros frames do vídeo, e a partir do momento em que o objeto é detectado utilizando unicamente as informações do modelo, o objeto passa a ser procurado por um rastreador que busca sempre o alvo nas possíveis transições alcançáveis a partir do aspecto atual identificado.

De maneira resumida, o modelo é utilizado como entrada para a criação de um detector e de um rastreador, sendo que o detector busca o objeto modelado, e a partir do momento em que o encontra, a localização é passada para o rastreador que busca o objeto nos possíveis estados alcançáveis a partir do estado atual. A Figura 6.2 mostra a estrutura dessa aplicação, desde a entrada inicial de um conjunto de frames utilizados para criar o modelo, até a saída que é a avaliação realizada em um novo conjunto de frames, resultando nas localizações obtidas através do modelo, e nas imagens mentais correspondentes a cada aspecto identificado por frame.

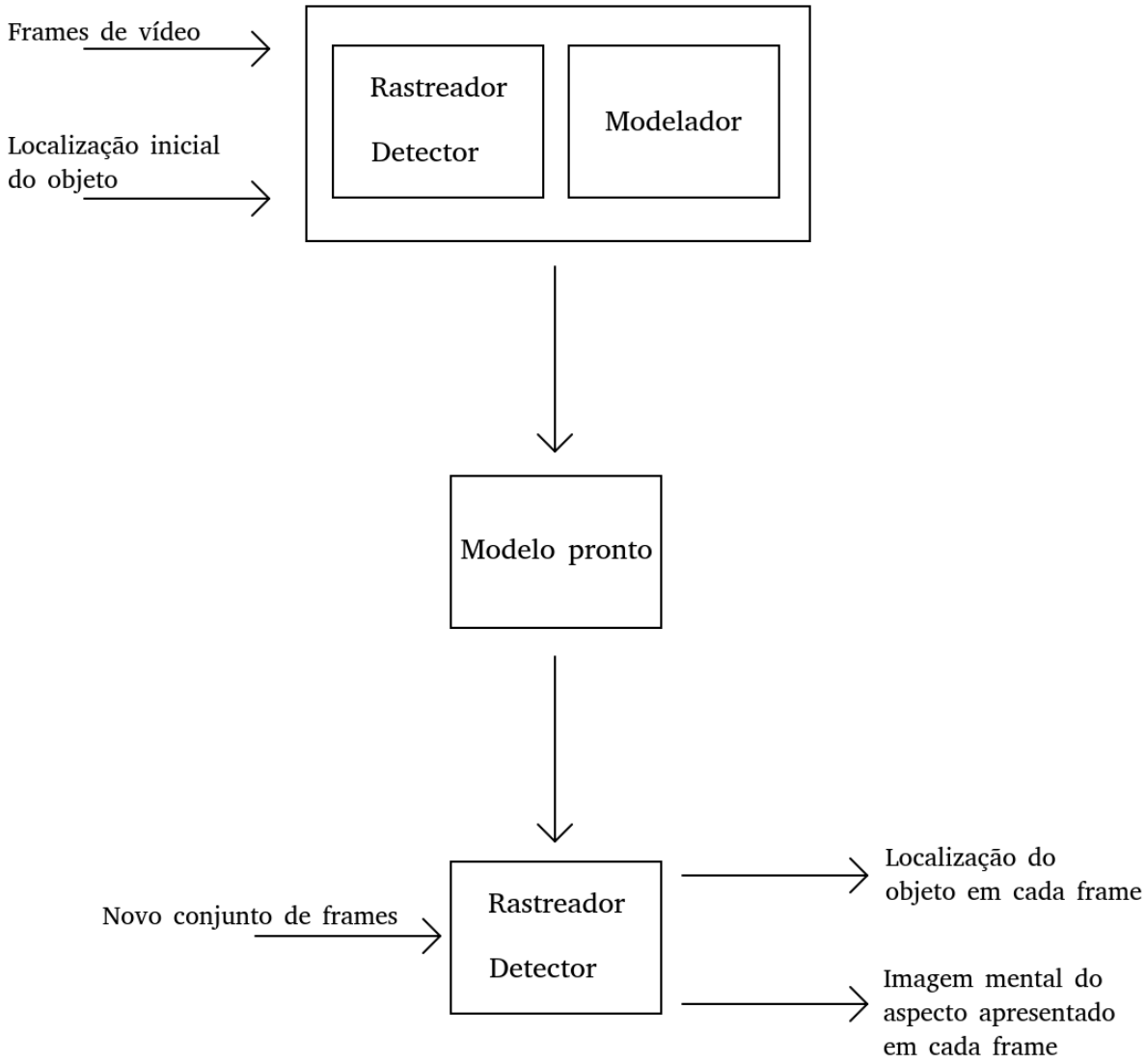

Figura 6.2: Rastreamento através de modelo. A modelagem do objeto é feita em um conjunto de frames, a partir da localização inicial no primeiro frame. Com o modelo pronto, este é avaliado em um novo conjunto dde frames. Para esta avaliação, cria-se um rastreador e um detector baseados somente nas informações do modelo, e para cada frame, a localização do objeto é determinada em conjunto com a imagem mental representante do aspecto identificado.

Os algoritmos 7, 8 e 9 a seguir, mostram como o rastreamento de um objeto é realizado partindo-se somente das informações apresentadas pelo seu modelo mental criado.

Algoritmo 7 Rastreamento através de modelo

Entrada: Modelo de um objeto com seus aspectos e transições entre aspectos
Entrada: Sequência de frames contendo o objeto alvo
  criaDetector(*modeloDeObjeto*)
  criaRastreador(*modeloDeObjeto*)
  para cada frame de vídeo **faça**
    Executa thread de rastreamento através de modelo
    Executa thread de detector através de modelo
    se *ObjetoDetectado* **então**
      se $CoordenadasDetector \neq CoordenadasRastreador$ **então**
        Reinicia rastreador através das coordenadas do detector com aspecto correspondente
      fim se
    fim se
    Apresenta imagem mental do aspecto identificado
  fim para

Algoritmo 8 Thread de rastreamento através de modelo

Entrada: Modelo de um objeto com seus aspectos e transições entre aspectos
Entrada: Frame atual
Entrada: Última localização identificada
Entrada: Último aspecto identificado
  para cada *aspecto* alcançável a partir do estado atual **faça**
    Procura o *aspecto* na vizinhança da última localização retornada
  fim para
  Atualiza aspecto atual
  Atualiza localização atual

Algoritmo 9 Thread de detecção através de modelo

Entrada: Modelo de um objeto com seus aspectos e transições entre aspectos

Entrada: Frame atual em escala reduzida

para cada *aspecto* pertencente ao modelo **faça**

Procura *aspecto* no frame reduzido

fim para

Atualiza aspecto atual

Atualiza localização atual

6.3.2 Resultados obtidos pelo rastreamento através de modelo

O rastreamento através dos modelos dos objetos foi realizado seguindo a abordagem apresentada na seção anterior, utilizando o dataset criado para esta finalidade [57]. A execução da criação dos modelos de objetos em tempo real, seguida da detecção e rastreamento realizados a partir dos modelos, pode ser visualizada em uma série de gravações disponibilizadas em https://link.springer.com/article/10.1007/s00521-024-09601-5, como material suplementar do artigo gerador desta tese, Object modeling through weightless tracking [33].

Em seguida, são apresentadas imagens de uma série de frames processados com os resultados obtidos, onde o rastreamento através dos modelos foi realizado, mostrando a localização encontrada para o alvo, juntamente com a imagem mental identificada pelo sistema em cada um dos frames. Nesta implementação, o modelo foi utilizado exclusivamente com as suas informações obtidas pelo aprendizado inicial, sem utilizar mais nenhum tipo aprendizado e atualização do modelo durante a execução, com o rastreamento sendo realizado através das buscas nas transições modeladas a partir de cada estado identificado.

Figura 6.3: Rastreamento através de modelo - Rosto

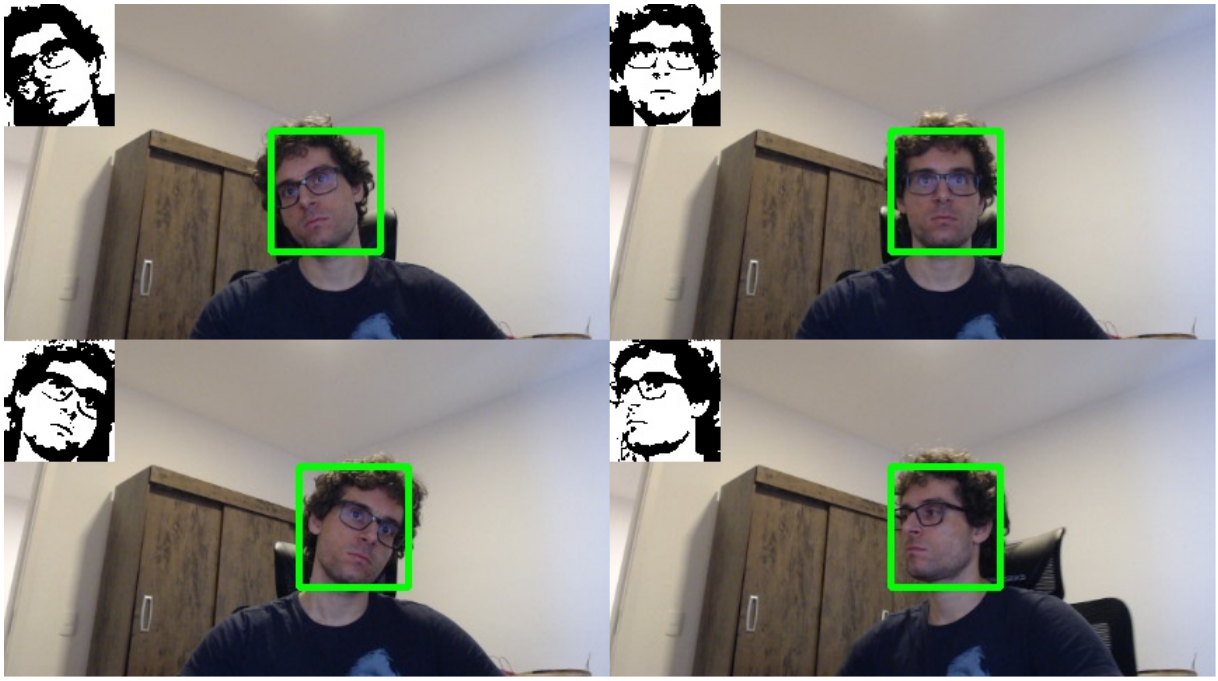

Figura 6.4: Rastreamento através de modelo - Rosto 2

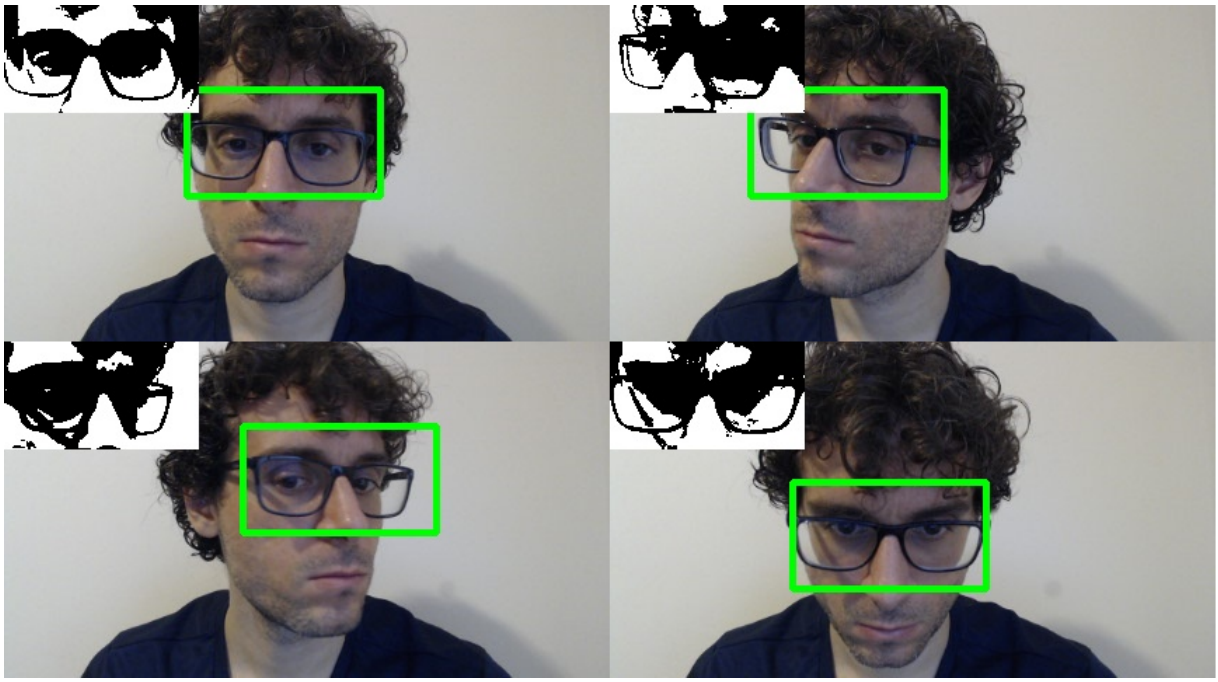

Figura 6.5: Rastreamento através de modelo - Óculos

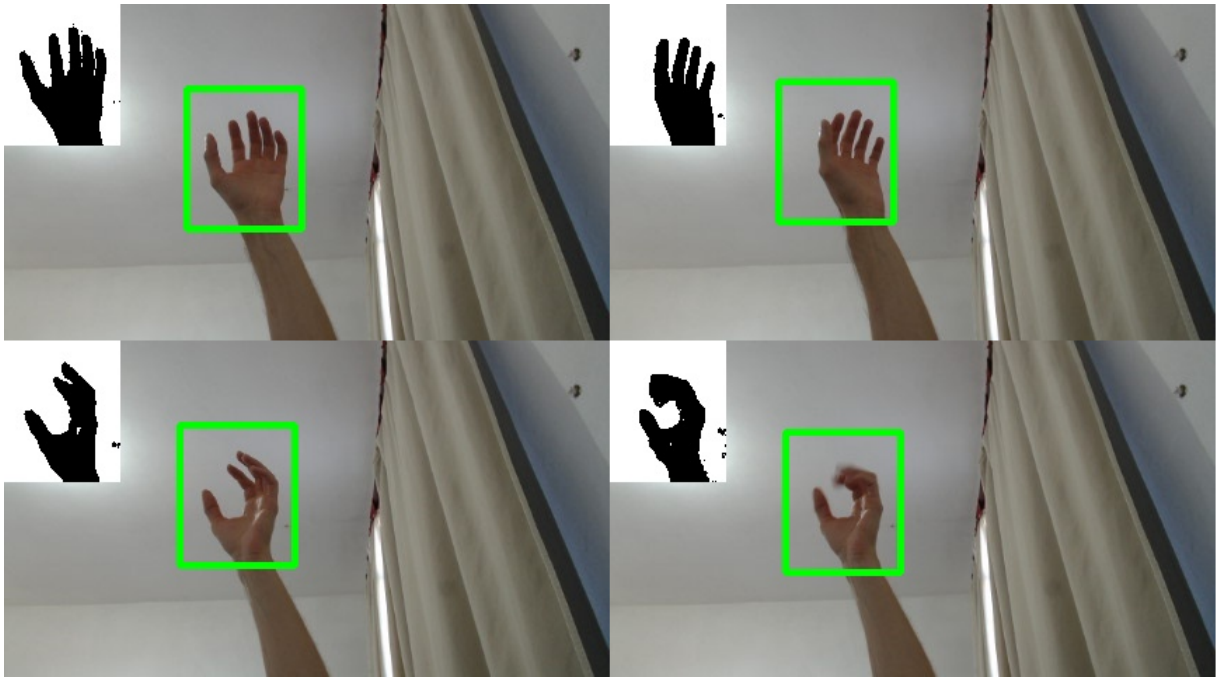

Figura 6.6: Rastreamento através de modelo - Mão

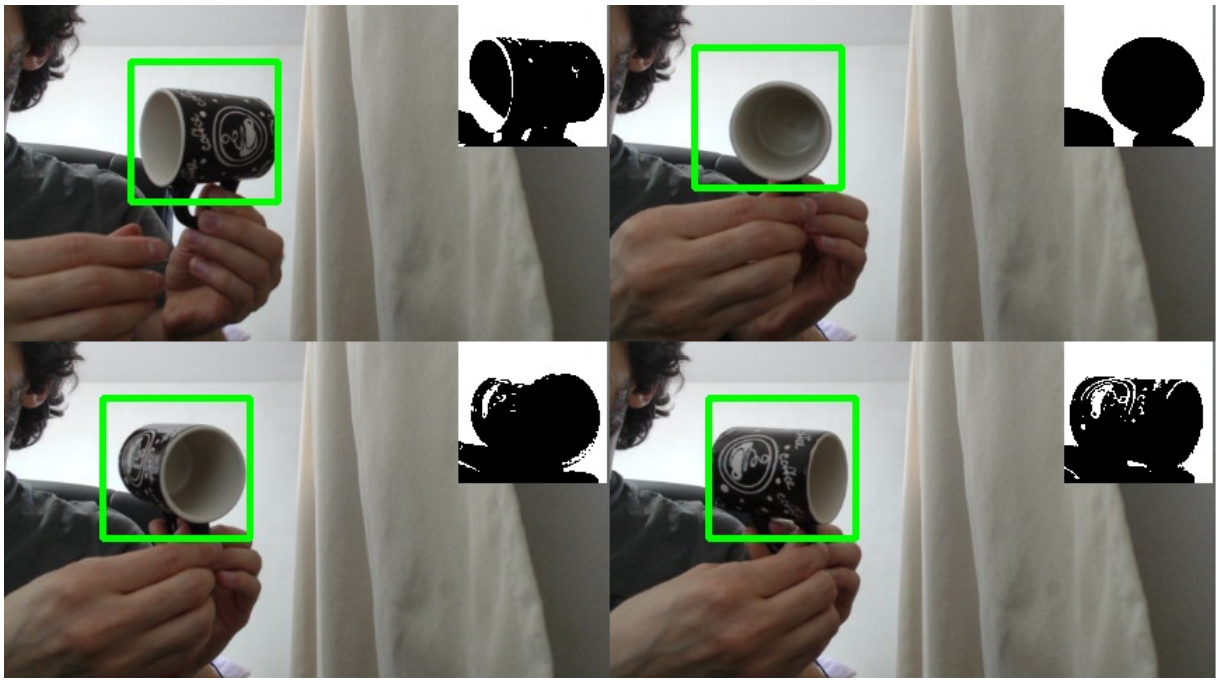

Figura 6.7: Rastreamento através de modelo - Xícara

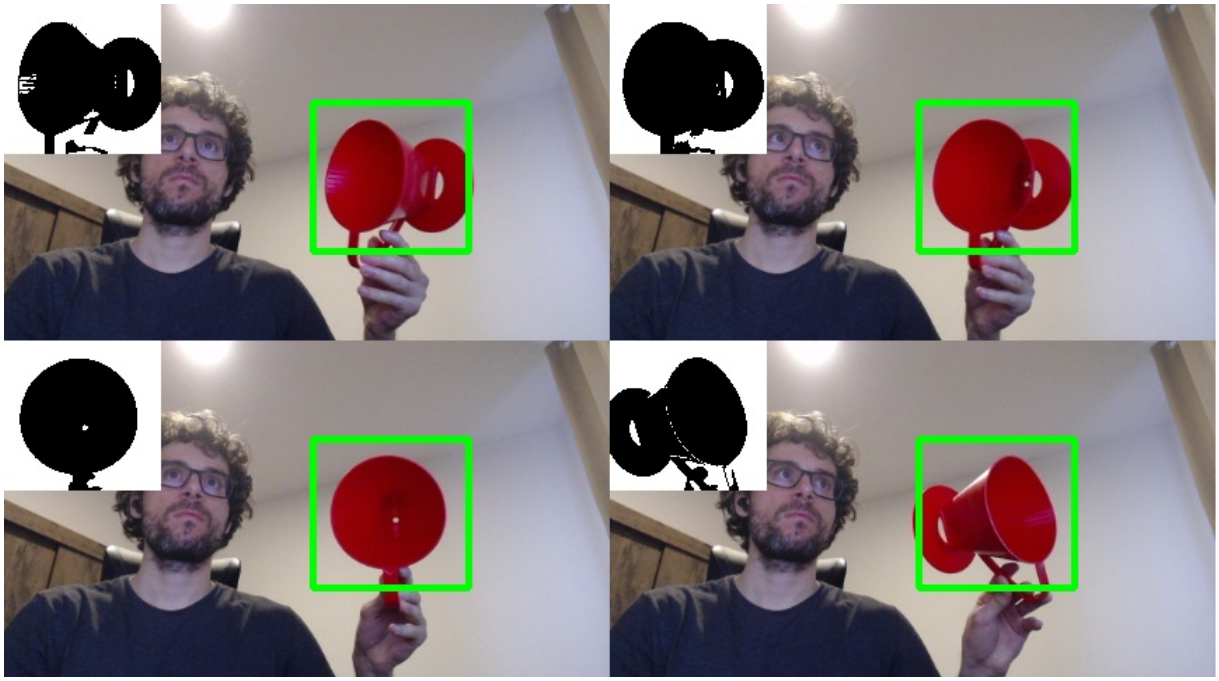

Figura 6.8: Rastreamento através de modelo - Coador de café

Figura 6.9: Rastreamento através de modelo - Fita adesiva

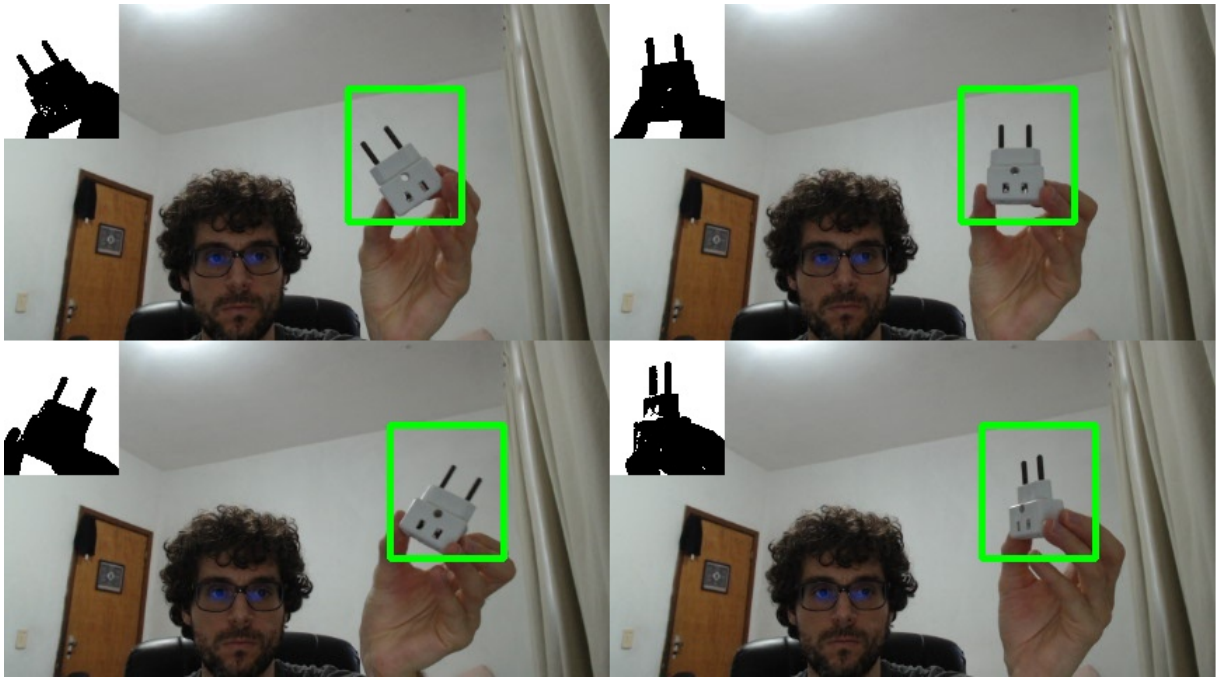

Figura 6.10: Rastreamento através de modelo - Adaptador de tomada

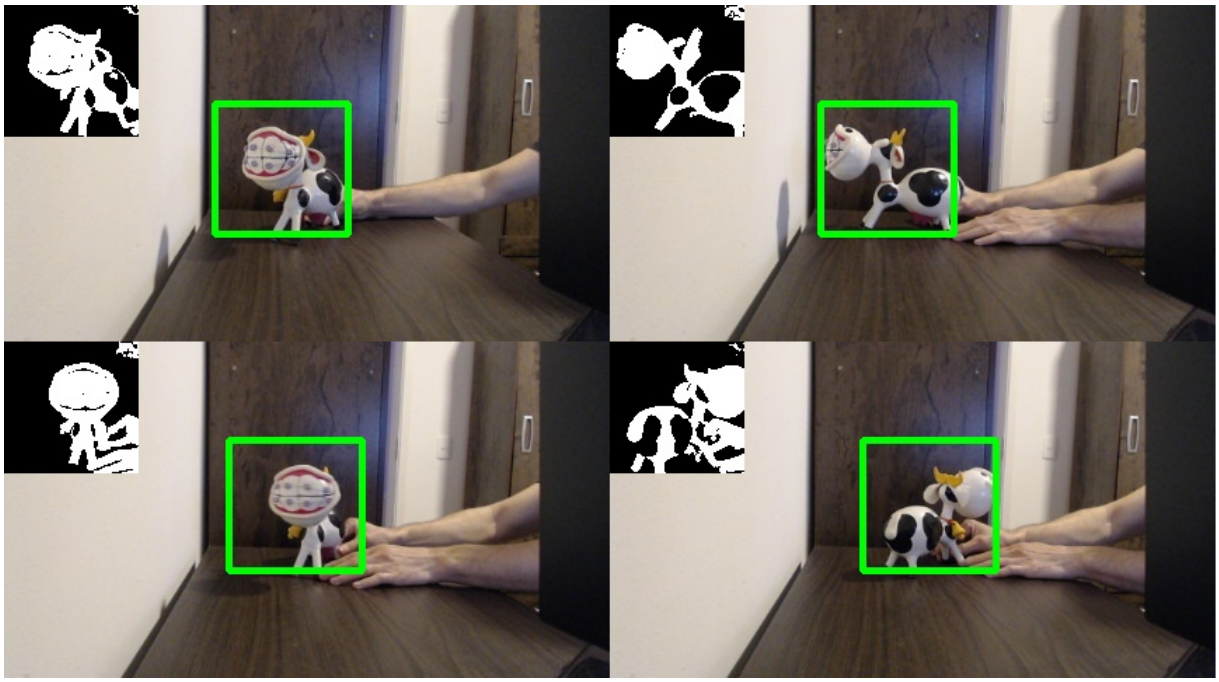

Figura 6.11: Rastreamento através de modelo - Vaca

Figura 6.12: Rastreamento através de modelo - Girafa

Como pode ser observado através dos resultados apresentados, os modelos conseguem produzir representações dos aspectos que fazem sentido visualmente. Para obter uma avaliação a partir de uma métrica mais precisa, foi utilizada a medida de similaridade entre os pixels, apresentada na Subseção 6.2.1, onde realiza-se a binarização do objeto em cada um dos frames do vídeo e realiza-se uma comparação pixel a pixel com a imagem mental retornada pelo sistema no frame correspondente. Dessa forma, cada frame original é convertido em tons de cinza e a média das intensidades dos pixels correspondentes à localização do objeto em cada frame é utilizada como limiar para determinar o novo valor dos pixels, onde pixels com valores acima deste limiar são convertidos para pixels brancos, enquanto pixels abaixo desse valor de limiar são convertidos para pixels pretos. Tendo a imagem binarizada do objeto no frame considerado, o sistema busca a imagem mental obtida pelo rastreamento através de modelo, descrito na seção anterior, e realiza uma comparação pixel a pixel para determinar a similaridade entre o objeto e a imagem mental obtida. As taxas médias de acerto de pixels por frame, obtidas em cada um dos vídeos analisados, são apresentadas na tabela a seguir.

Vídeo	**Taxa média de acerto de pixels por frame**
Rosto	0.83
Cabeça	0.89
Óculos	0.89
Mão	0.94
Xícara de café	0.93
Coador de café	0.92
Fita adesiva	0.87
Adaptador de tomada	0.93
Vaca	0.85
Girafa	0.90

Tabela 6.1: Resultados de taxa de acerto de pixels

Os resultados obtidos de imagens mentais para cada um dos aspectos observados nos frames analisados foram satisfatórios quando avaliados qualitativamente, de acordo com as representações visuais apresentadas, e a métrica de similaridade entre os objetos apresentados nos frames e as representações dos aspectos determinadas pelos modelos, corroboram que este método de modelagem através de imagens mentais é uma boa representação para mapear objetos do mundo real para serem utilizados como informações em sistemas computacionais.

Nos experimentos realizados, tanto a modelagem quanto o rastreamento são executados em tempo real, e foi possível adicionar a funcionalidade de modelagem de

objetos ao rastreamento de objetos, mantendo uma boa performance em frames processados por segundo, como mostram os resultados na tabela abaixo, obtidos em um notebook Intel Core i7-8750H com 16 GB de memória RAM.

Vídeo	**FPS - Rastreamento**	**FPS - Rastreamento e modelagem**
Rosto	49,48	48,93
Cabeça	51,56	50,27
Óculos	60,12	59,09
Mão	59,87	59,08
Xícara de café	66,31	65,23
Coador de café	68,19	66,63
Fita adesiva	54,01	51,99
Adaptador de tomada	35,02	34,52
Vaca	59,49	58,69
Girafa	60,25	57,98

Tabela 6.2: Medidas de frames por segundo

6.3.3 Utilização de modelos em diferentes escalas

Esta seção apresenta os experimentos realizados com o objetivo de verificar a aplicabilidade de um único modelo de objeto treinado sendo utilizado para rastrear diferentes escalas. Deste modo, os modelos prontos foram redimensionados através das imagens mentais, ou seja, cada uma das imagens mentais pertencentes ao modelo foi redimensionada para uma nova escala. Estas imagens mentais redimensionadas são utilizadas para treinar novos discriminadores em tamanhos correspondentes às novas escalas. As transições entre os aspectos redimensionados são as mesmas dos modelos originais.

A quantidade de RAMs de um discriminador é determinada em função da quantidade de pixels presentes dentro do bounding box delimitador do objeto e do número de bits utilizados, ou seja, sendo b a quantidade de bits e n a quantidade de pixels dentro do bounding box do objeto, então, a quantidade de RAMs do discriminador é dada por n/b. Como o número de bits utilizado para criar os discriminadores redimensionados é mantido fixo, então, um discriminador treinado a partir de uma imagem mental redimensionada possui uma quantidade de RAMs diferente da quantidade de RAMs do discrminador original. O processo de criação de discriminadores para representar escalas diferentes de um aspecto de um mesmo objeto é ilustrado na Figura 6.13, e em seguida, a Figura 6.14 ilustra a obtenção de modelos mentais redimensionados para representar um mesmo objeto em diferentes escalas.

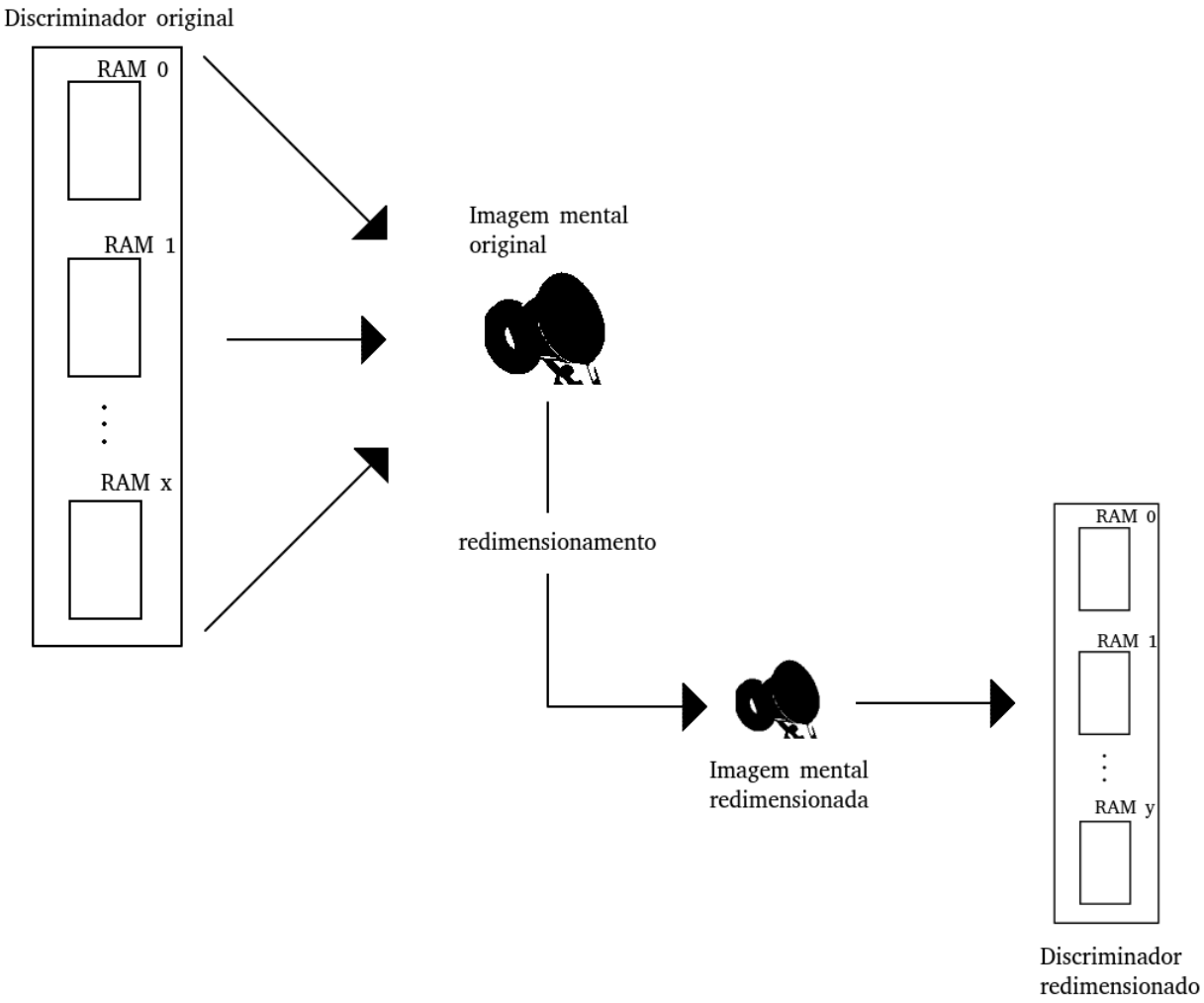

Figura 6.13: Redimensionamento de discriminador a partir de imagem mental. Um discriminador treinado retorna uma imagem mental de um determinado aspecto. Essa imagem mental é redimensionada e utilizada para treinar um novo discriminador. Dessa forma, esse novo discriminador é capaz de reconhecer o mesmo aspecto em tamanho redimensionado.

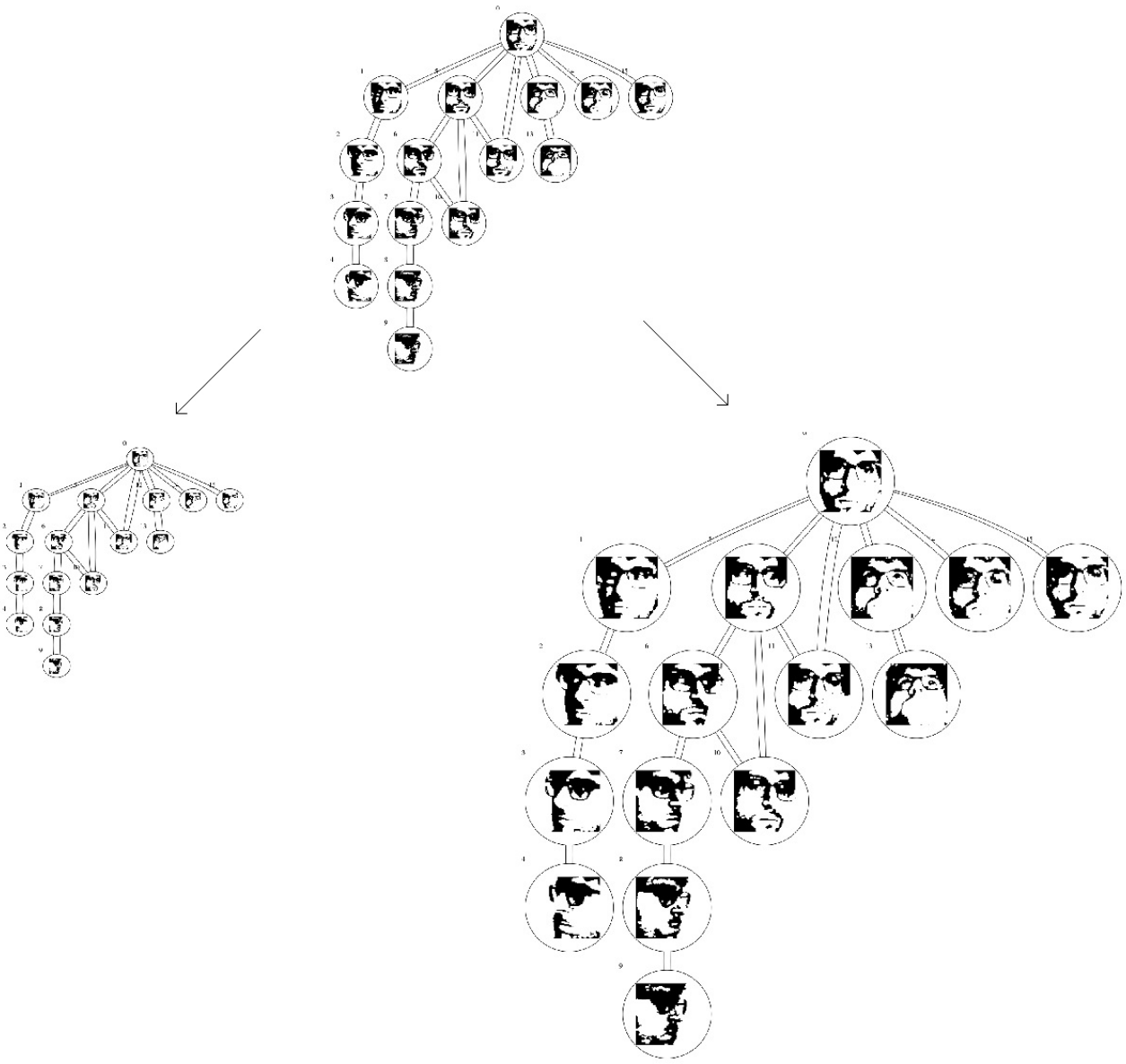

Figura 6.14: Redimensionamento de modelo. Um modelo mental, formado por imagens mentais de aspectos e suas transições, redimensionado para diferentes escalas. Estes novos modelos são utilizados para treinar novos discriminadores que reconhecem o objeto mapeado em diferentes tamanhos. As transições entre aspectos são as mesmas em todos os modelos.

O mesmo método de rastreamento através de modelo foi utilizado para avaliar a utilização dos modelos aplicados a diferentes escalas. Dessa forma, um modelo para representar um objeto é treinado em um conjunto de frames e avaliado em um outro conjunto de frames, onde ocorrem mudanças de escala do objeto. A identificação dessas mudanças de escala são realizadas através das buscas nos modelos mentais redimensionados, onde foram utilizadas duas escalas de redimensionamento, uma para ampliar os aspectos dos modelos originais e a outra para reduzí-los. A tabela a seguir mostra os resultados de taxa de acerto de pixels por frame, aplicando a mesma métrica de similaridade entre pixels apresentada na seção anterior, e em seguida podem ser observados alguns resultados do processamento realizado nos frames redimensionados, mostrando a localização do alvo e o aspecto correspondente identificado pelo rastreamento através de modelo.

Vídeo	**Modelo em escala reduzida**	**Modelo em escala ampliada**
Rosto	0,82	0,85
Xícara de café	0,85	0,81
Fita adesiva	0,82	0,85
Adaptador de tomada	0,88	0,88
Vaca	0,87	0,81

Tabela 6.3: Taxa de acerto média de pixels por frame utilizando modelos redimensionados

Figura 6.15: Rastreamento em escalas diferentes de um mesmo modelo - Rosto

Figura 6.16: Rastreamento em escalas diferentes de um mesmo modelo - Xícara

Figura 6.17: Rastreamento em escalas diferentes de um mesmo modelo - Fita adesiva

Figura 6.18: Rastreamento em escalas diferentes de um mesmo modelo - Adaptador de tomada

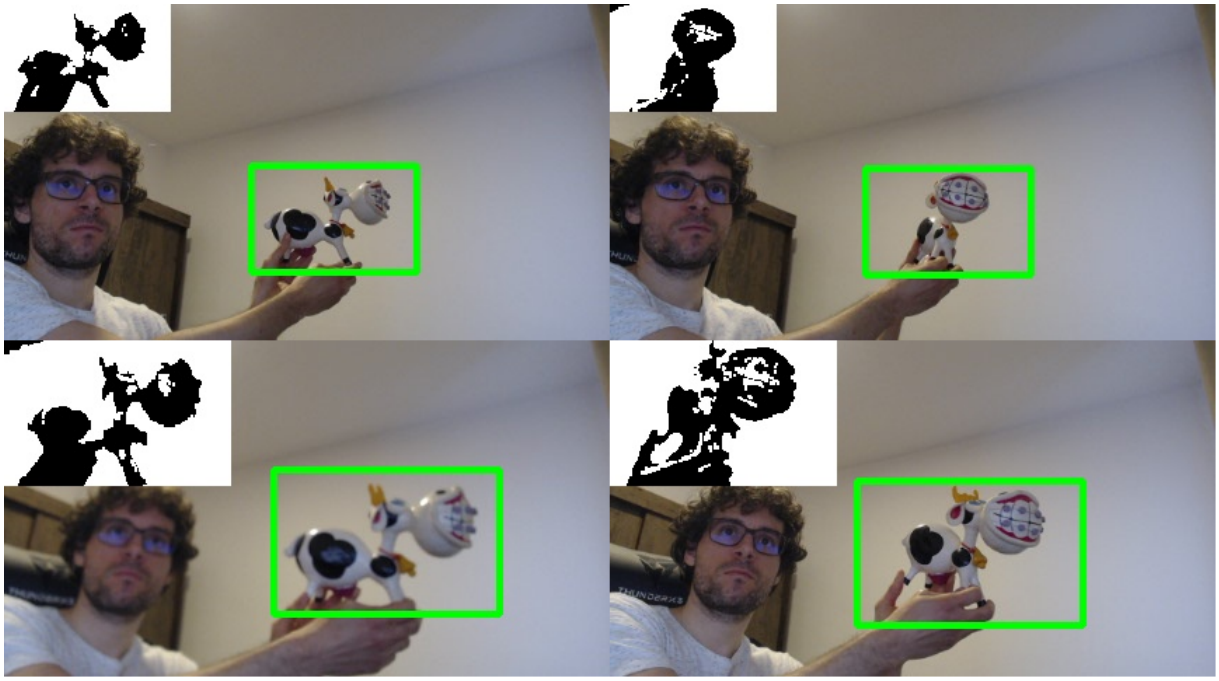

Figura 6.19: Rastreamento em escalas diferentes de um mesmo modelo - Vaca

Os resultados obtidos mostram que foi possível utilizar um mesmo modelo para obter modelos redimensionados com discriminadores utilizados para representar os aspectos em diferentes tamanhos. É interessante observar que desta forma, a partir de um único modelo de discriminadores treinado, foi possível classificar padrões de entrada de tamanhos diferentes, sem a necessidade de adaptar os tamanhos das entradas, pois o classificador foi modificado para aceitar entradas de tamanhos diferentes e não ao contrário, com a padronização dos tamanhos das entradas a serem classificadas, como seria uma abordagem convencional na classificação de padrões de tamanhos diferentes.

6.4 Avaliação das transições entre aspectos do modelo

O objetivo desta seção é avaliar se as transições entre aspectos geradas nos modelos mentais são adequadas. Desta forma, o problema de motion planning [60] foi considerado, onde objetos em um determinado estado inicial são manipulados por robôs com o objetivo de alcançar um estado final. Para avaliar as transições geradas, foi criada uma aplicação que recebe um modelo pronto e duas imagens como entrada, uma representando o estado inicial do objeto e a outra representando o estado final a ser alcançado. Desta forma, o passo inicial é classificar as imagens de estado inicial e final nos discriminadores do modelo, a fim de determinar quais discriminadores são mais semelhantes com os essas imagens de entrada. Após a determinação de discriminadores mais semelhantes, a partir do estado inicial, realiza-se uma busca no grafo de estados dos modelos, seguindo as possíveis transições mapeadas, a fim de alcançar o estado final desejado. Pode-se então representar os possíveis caminhos de estados entre dois aspectos de um mesmo objeto, obtendo uma avaliação visual sobre as transições realizadas nas aparências dos modelos a fim de movimentar um objeto de um aspecto inicial até um aspecto destino. Alguns resultados de caminhos de estados obtidos são apresentados nas sequências de transições entre imagens mentais abaixo. As imagens coloridas representam os estados iniciais e finais dados como entrada para a aplicação, e as imagens do caminho de estados são as imagens mentais obtidas dos discriminadores dos modelos.

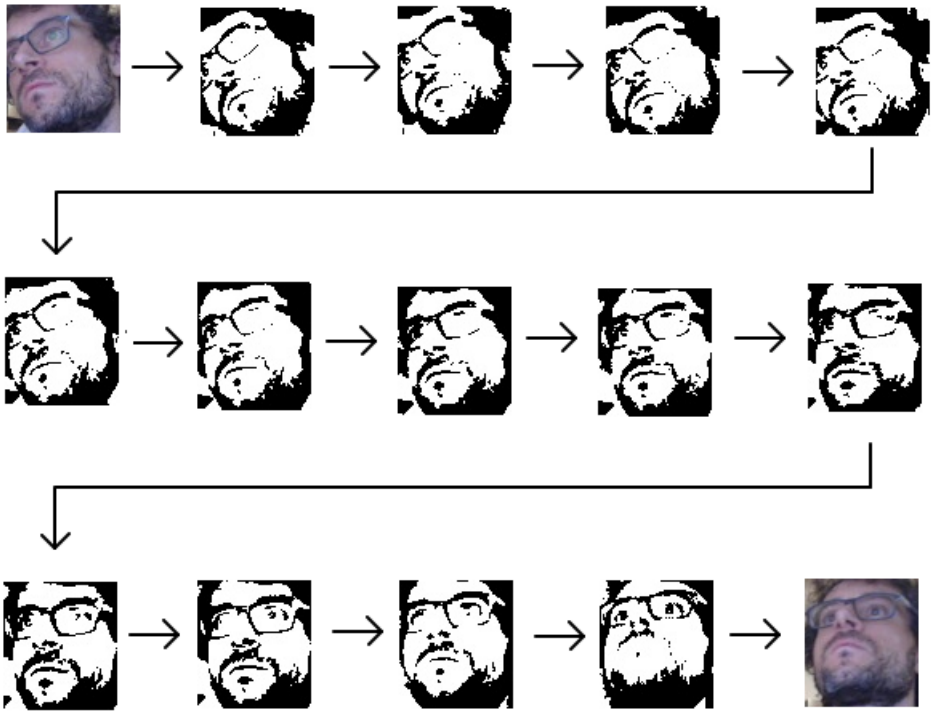

Figura 6.20: Caminho de estados - Rosto

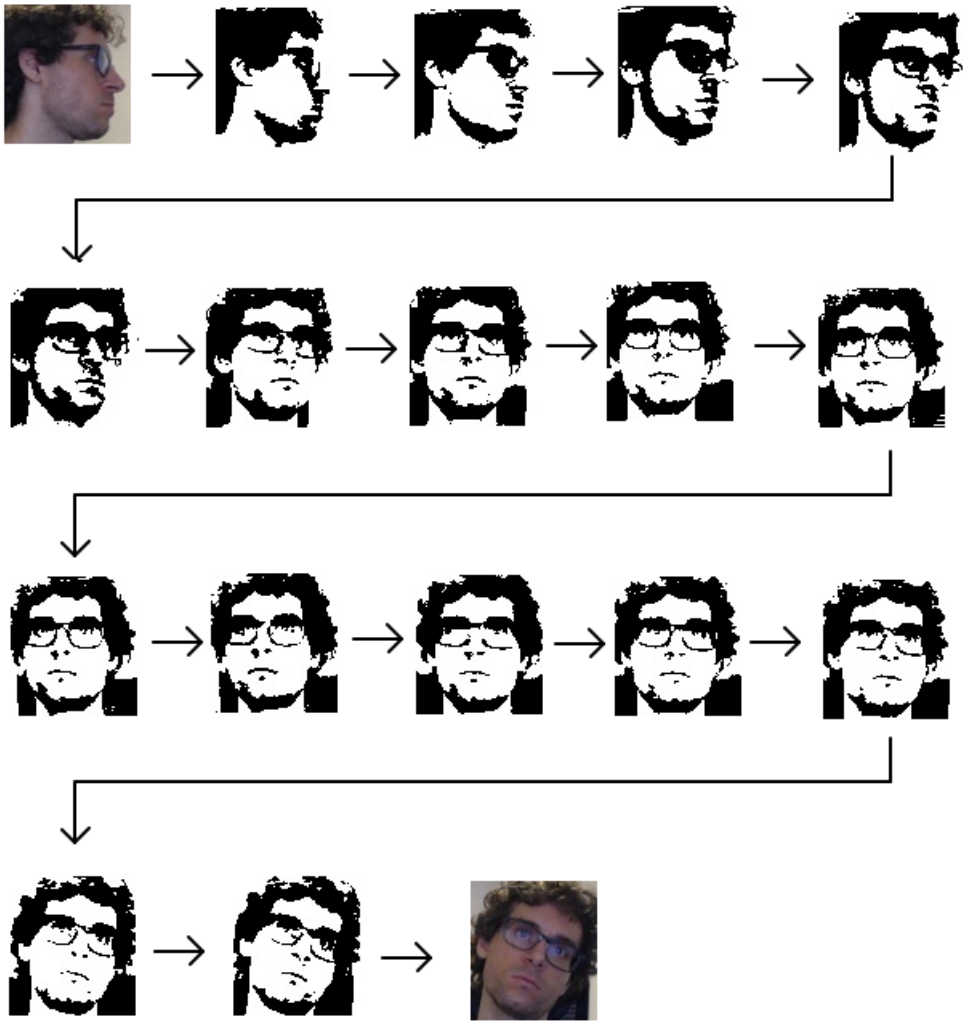

Figura 6.21: Caminho de estados - Cabeça

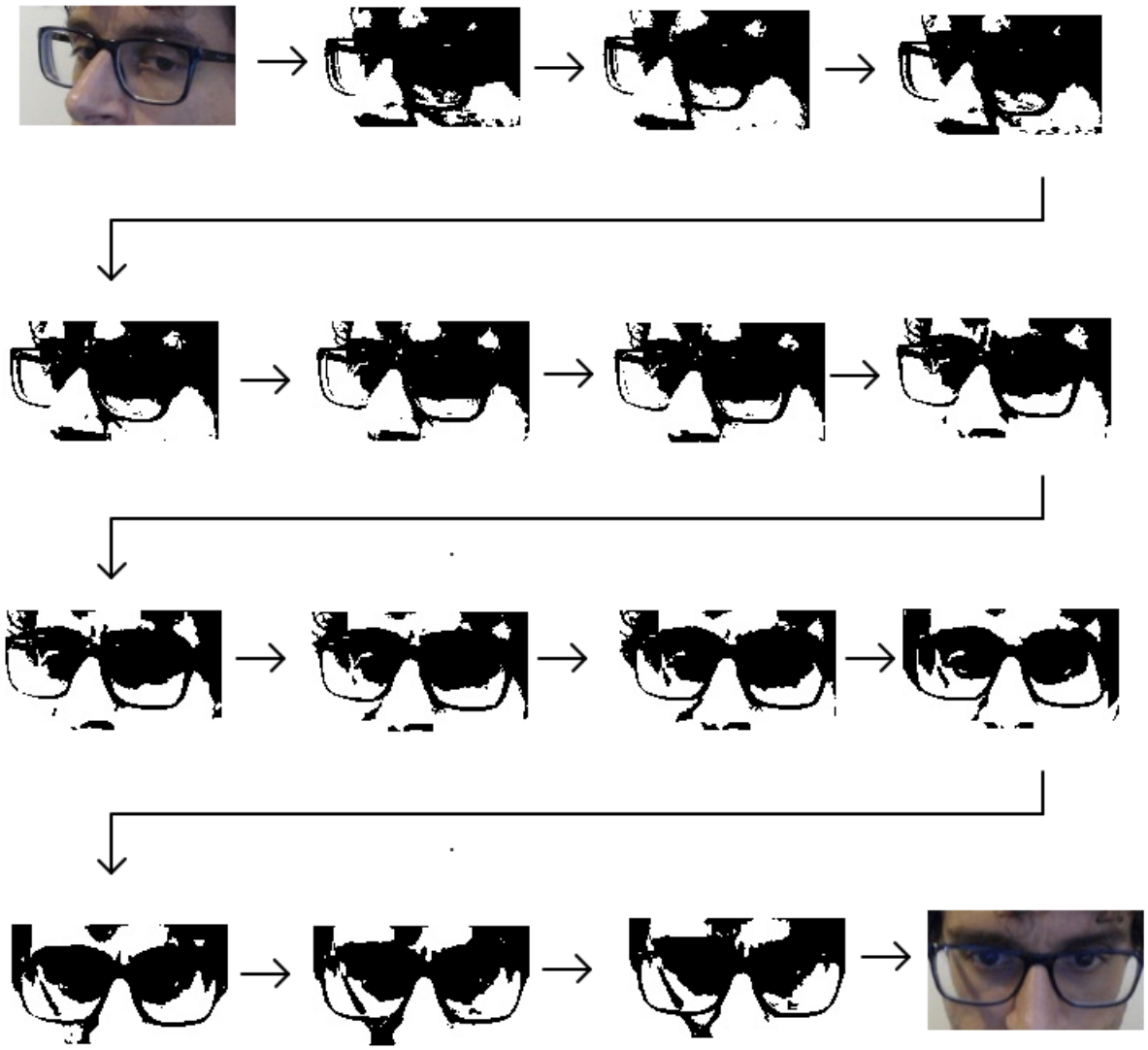

Figura 6.22: Caminho de estados - Óculos

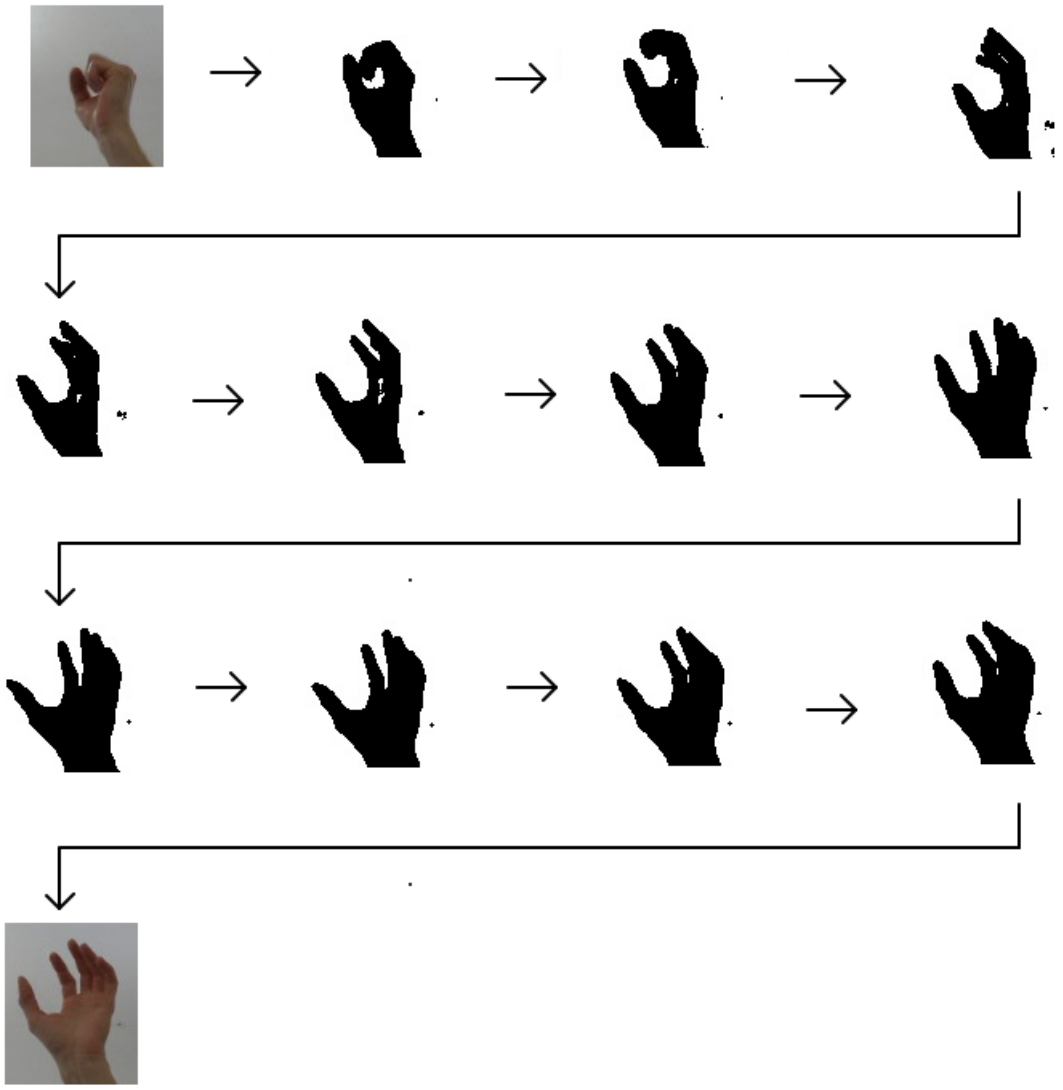

Figura 6.23: Caminho de estados - Mão

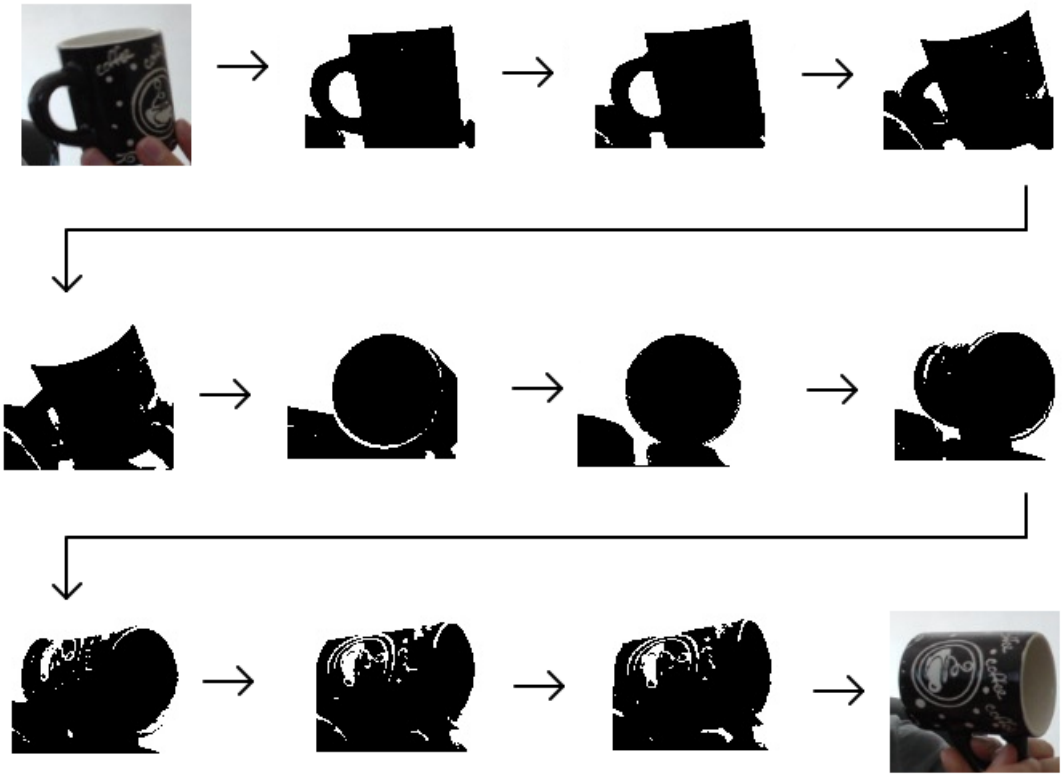

Figura 6.24: Caminho de estados - Xícara

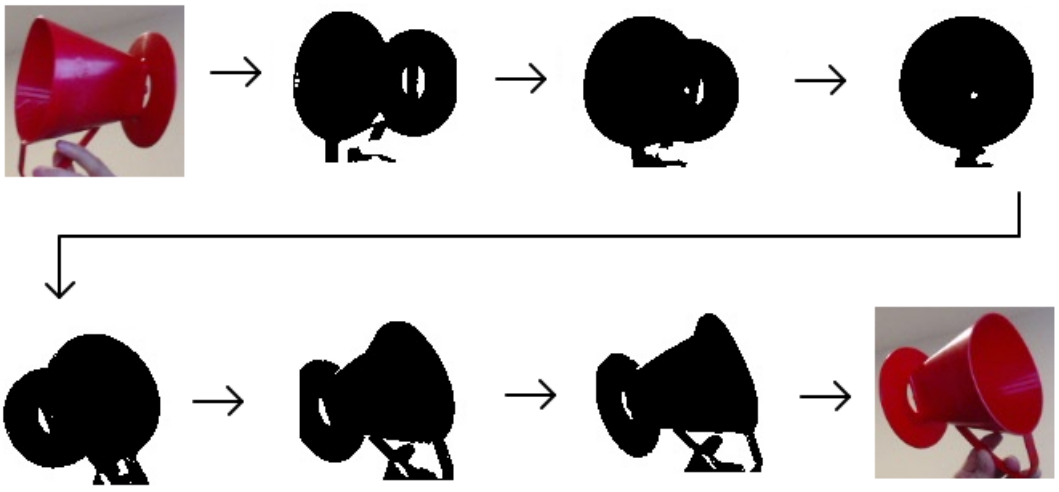

Figura 6.25: Caminho de estados - Coador de café

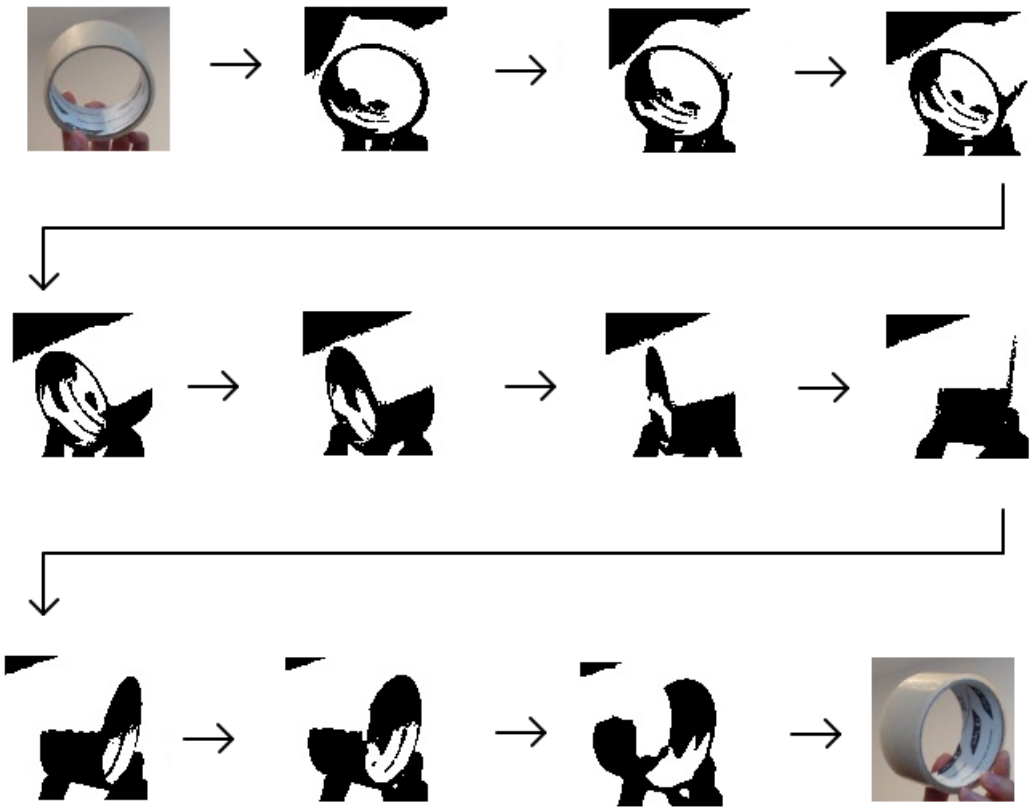

Figura 6.26: Caminho de estados - Fita adesiva

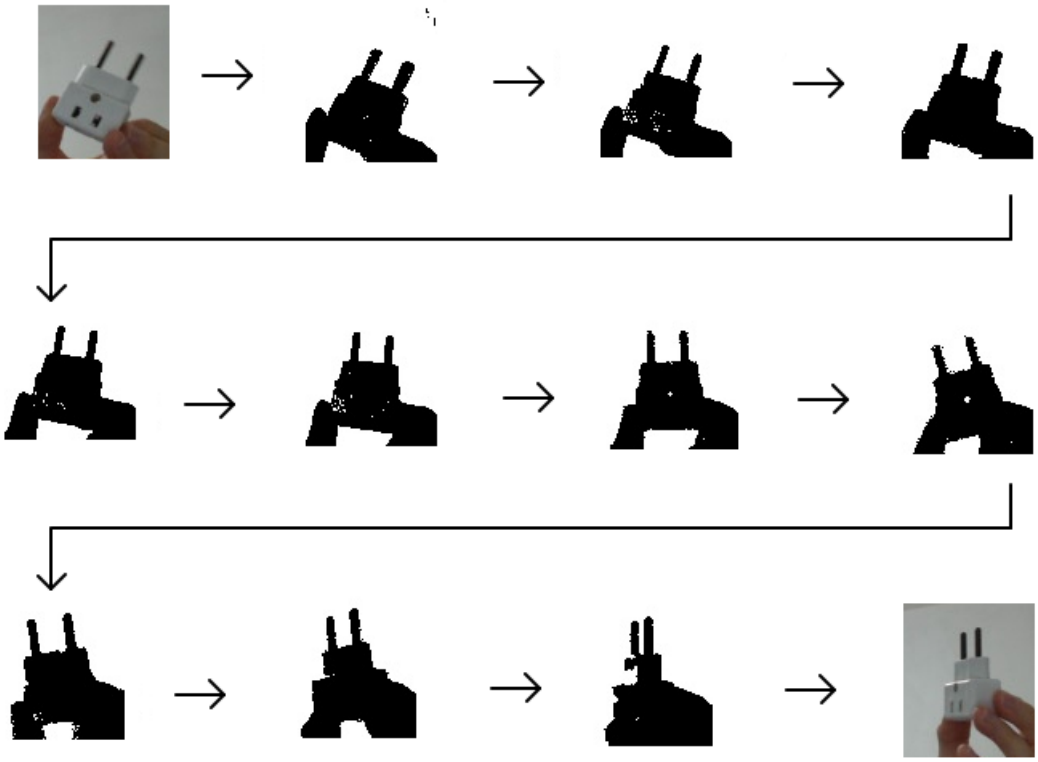

Figura 6.27: Caminho de estados - Adaptador de tomada

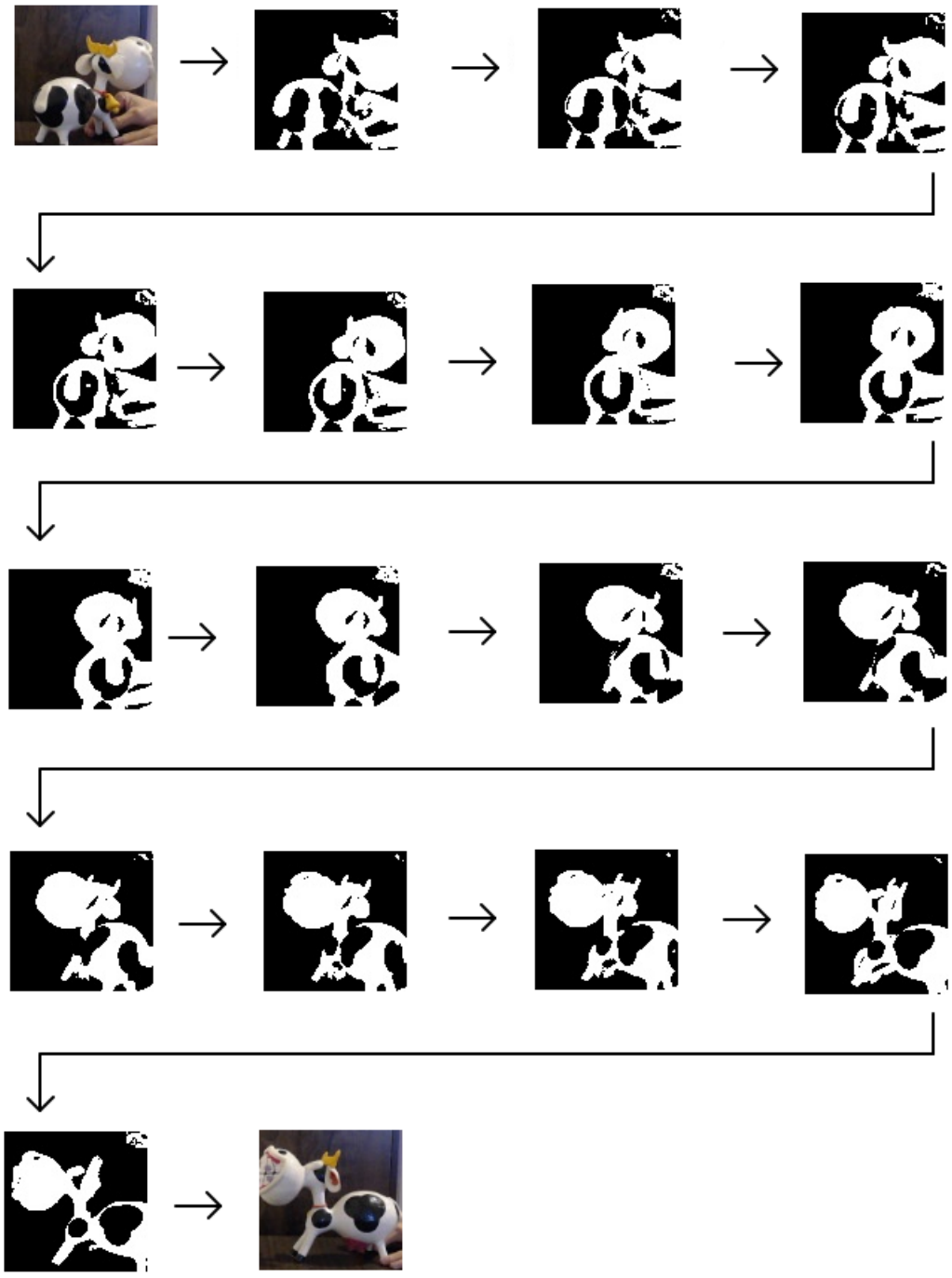

Figura 6.28: Caminho de estados - Vaca

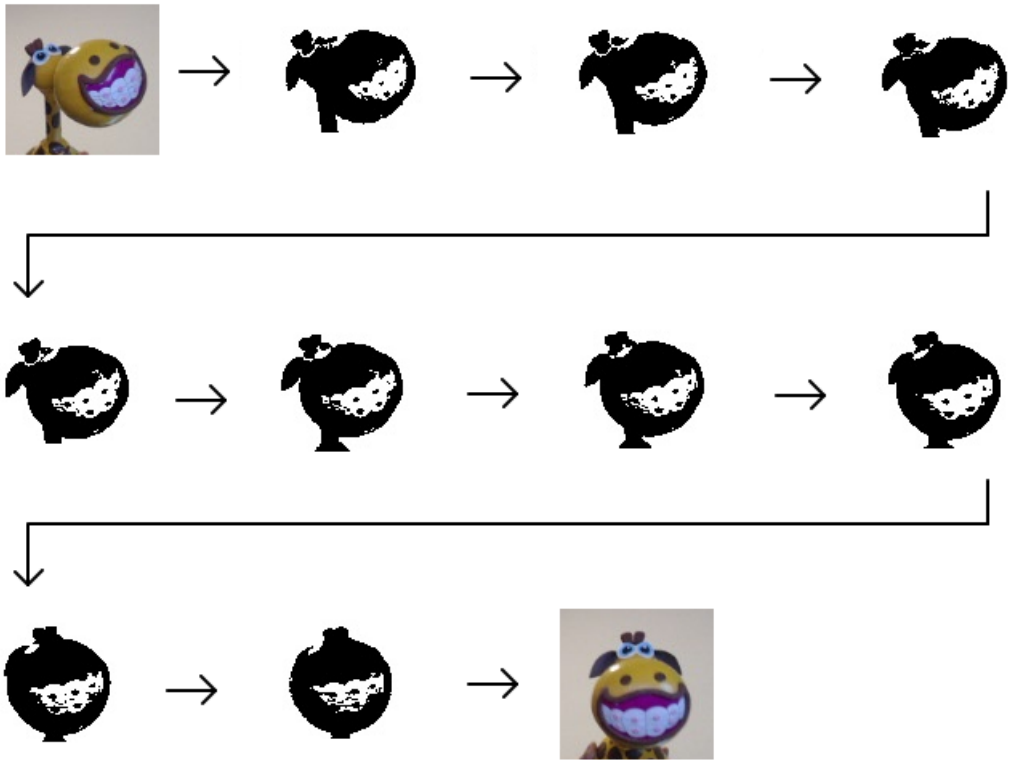

Figura 6.29: Caminho de estados - Girafa

Os resultados apresentados mostram que o mapeamento de transições implementado na criação dos modelos foi adequado para representar as relações entre os diferentes aspectos de um mesmo objeto. Os resultados visuais mostram que as sequências de estados retornadas pelo sistema fornecem caminhos representantes de movimentações de objetos. Sendo assim, os modelos gerados podem fornecer realmente um entendimento sobre as formações visuais de um objeto observado.

6.5 Oclusão parcial

Esta seção apresenta a utilização de modelos para visualizar partes de um objeto que sofrem oclusão durante o rastreamento, de maneira que o sistema consegue prever como serão as apresentações dos aspectos parcialmente escondidos.

6.5.1 Identificação de oclusão parcial através do modelo

A solução encontrada para visualizar partes escondidas de um objeto utiliza a abordagem de subdiscriminadores apresentada na Seção 4.3. Desta forma, os modelos são criados com discriminadores formados por subdiscriminadores responsáveis por classificar partes do objeto, como exemplifica a Figura 6.30.

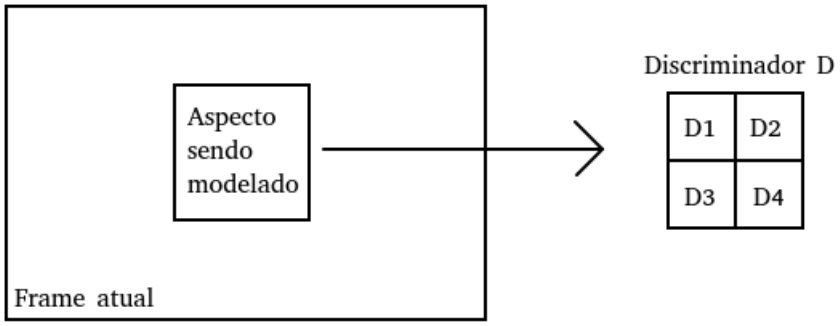

Figura 6.30: Aspecto com subdiscriminadores. Para identificar oclusões parciais, cada aspecto é modelado através de subdiscriminadores.

Sendo um aspecto formado por um discriminador contendo subdiscriminadores, assume-se que caso ocorra a situação onde existem subdiscriminadores retornando pontuações acima de um determinado limiar de aceitação, enquanto alguns dos subdiscriminadores retornam pontuações abaixo de um limiar de oclusão, então estes últimos representam as regiões do objeto que estão sofrendo oclusão.

6.5.2 Visualização de partes escondidas dos objetos

Para realizar a visualização de partes escondidas de um objeto, a classificação do aspecto é realizada nos subdiscriminadores dos discriminadores do modelo. Nesta

implementação, foram utilizados quatro subdiscriminadores para cada discriminador, e assim, para cada frame rastreado, cada parte do objeto localizado retorna a mais alta pontuação obtida por algum dos subdiscriminadores de aspectos existentes no modelo. Caso ocorra a situação de existência de pontuações acima de um limiar de aceitação e abaixo de um limiar de oclusão, simultaneamente, para diferentes subdiscriminadores de um mesmo aspecto, então, pode-se desenhar as imagens mentais das partes que estão escondidas, selecionando as partes das imagens mentais do aspecto identificado corretamente, e aplicando-as nas posições correspondentes às oclusões identificadas no frame avaliado. A figura a seguir exemplifica esta situação.

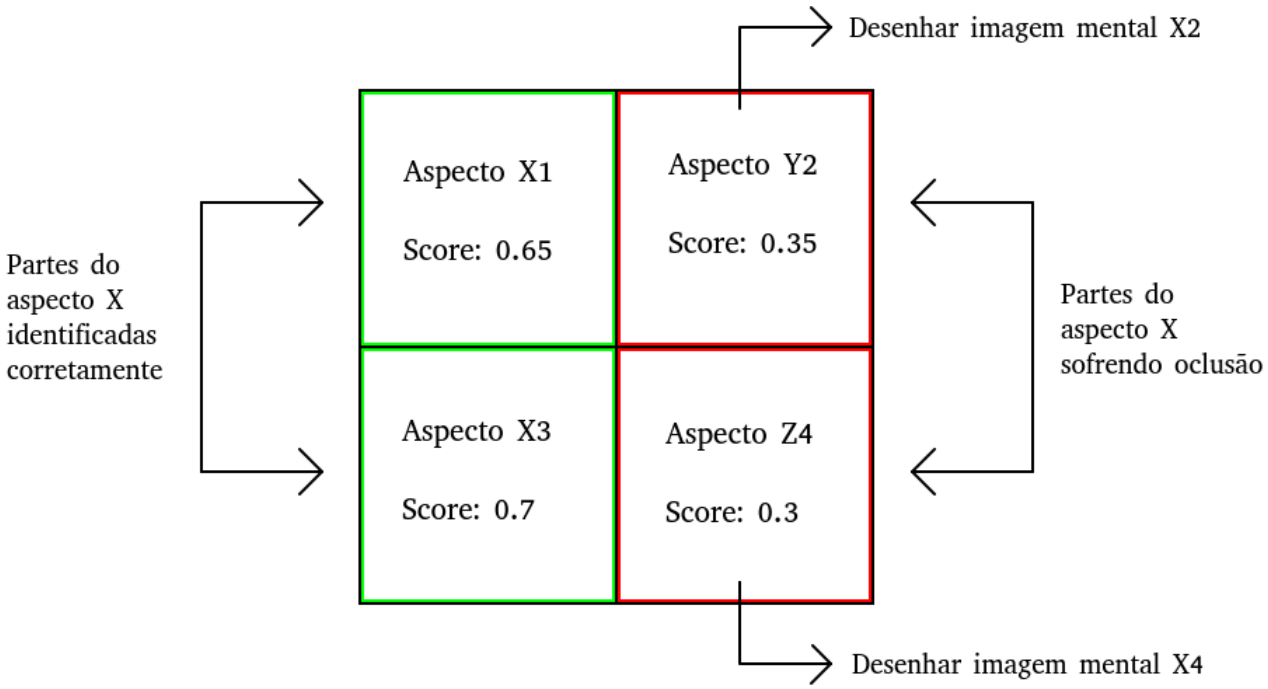

Figura 6.31: Visualização de aspectos escondidos. Neste exemplo, os subdiscriminadores X1 e X3 retornam pontuações acima de um limiar de aceitação, identificando as partes do aspecto X corretamente. Na parte direita da imagem, as pontuações obtidas foram abaixo de um limiar de oclusão, indicando que esses aspectos estão errados. Desta forma, as imagens mentais do lado direito do aspecto X devem ser desenhadas para obter uma visualização das partes sofrendo oclusão.

A execução da criação do modelo, seguida da utilização para visualizar partes escondidas dos objetos podem ser vistas em https://link.springer.com/article/10.1007/s00521-024-09601-5, no material suplementar do artigo Object modeling through weightless tracking [33]. As imagens a seguir mostram alguns frames processados com a visualização de partes escondidas

dos objetos.

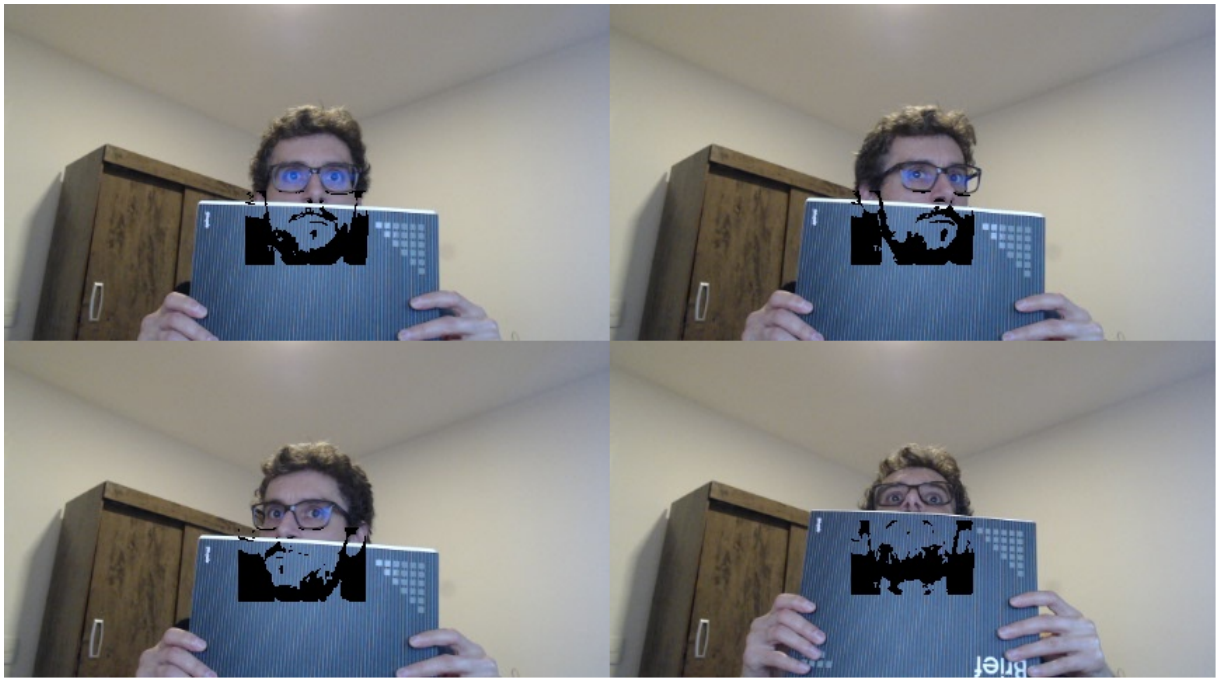

Figura 6.32: Visualização de oclusão parcial - Rosto

Figura 6.33: Visualização de oclusão parcial - Rosto 2

Figura 6.34: Visualização de oclusão parcial - Coador de Café

Figura 6.35: Visualização de oclusão parcial - Xícara

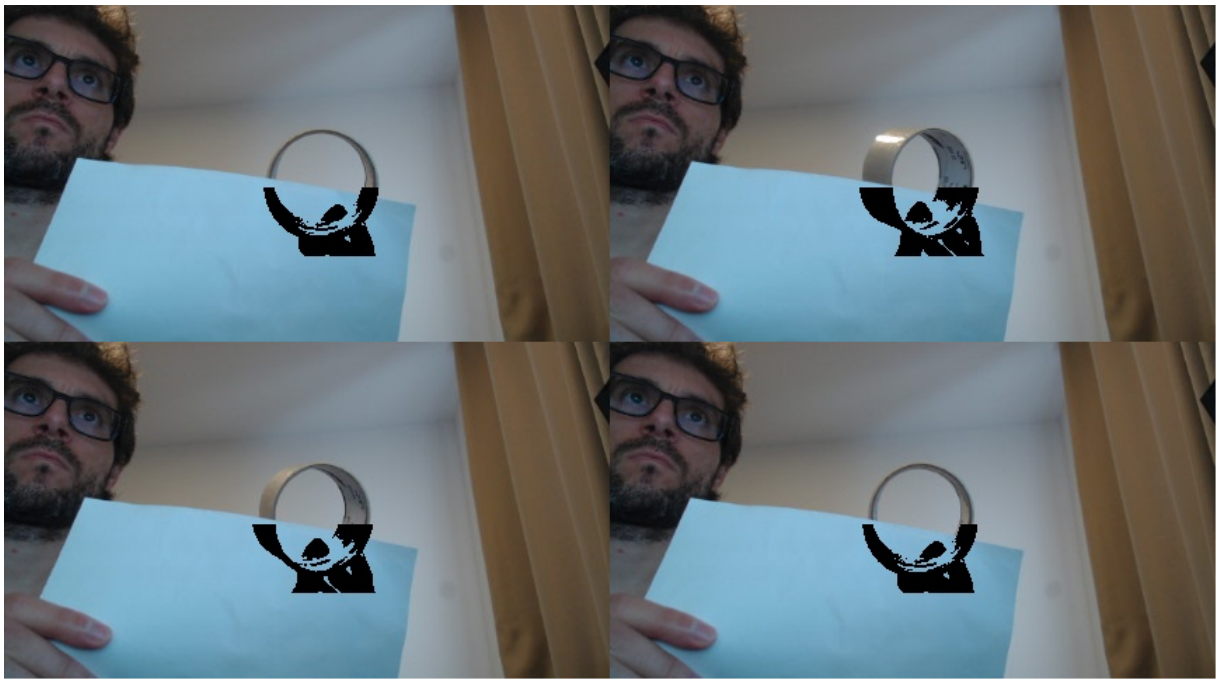

Figura 6.36: Visualização de oclusão parcial - Fita Adesiva

Figura 6.37: Visualização de oclusão parcial - Girafa

Os resultados visuais apresentados mostram que foi possível utilizar os modelos para identificar as partes escondidas dos objetos e apresentá-las na imagens, demonstrando que o sistema desenvolvido é capaz de utilizar os modelos como fornecedores de conhecimento sobre as informações visuais dos objetos, sendo possível prever como as partes sofrendo oclusão se apresentam nos vídeos observados.

6.6 Comparação com outros rastreadores

Existem diversos rastreadores que executam bem a tarefa de rastreamento, porém, sem um retorno de um modelo visual mapeando as transições entre aspectos. Desta forma, o objetivo desta seção é realizar uma comparação com outros rastreadores a fim de avaliar a capacidade de rastrear os objetos corretamente, mostrando que o sistema proposto é capaz de criar boas representações visuais, e também é capaz de realizar satisfatoriamente a tarefa do rastreamento quando aplicado em situações que forneçam variados tipos de dificuldade.

A avaliação foi realizada na base de dados OTB100 [58], através da métrica Intersection over Union (IoU) [59], apresentada na Subseção 6.2.2, onde em cada frame, é calculada a área de interseção entre o bounding box previsto e o bounding box representando a resposta correta da localização do alvo. Os resultados de rastreamento foram avaliados através das implementações disponíveis na biblioteca OpenCV [61] dos seguintes rastreadores: Boosting [27], MIL [62], KCF [29], CSRT [32], MedianFlow[22], TDL[63], MOSSE[30] e GOTURN[31].

A Tabela 6.4 apresenta os resultados de IoU médios para cada um dos rastreadores avaliados utilizando o dataset OTB100, juntamente com as taxas de acerto, onde considera-se que o rastreamento foi realizado de maneira correta em cada um dos frames quando são obtidos valores de IoU acima de 0,5.

Rastreador	**IoU médio**	**Taxa de acerto (limiar de 0,5)**
Boosting	0,3752	0,3997
MIL	0,3402	0,3410
KCF	0,2710	0,3166
CSRT	0,5225	0,5821
MedianFlow	0,2863	0,3163
TLD	0,3137	0,3370
MOSSE	0,2620	0,2711
GOTURN	0,1685	0,1112
WeightlessTracker	0,3852	0,4183

Tabela 6.4: Acurácia dos rastreadores no dataset OTB-100

Como complemento da Tabela 6.4, o gráfico a seguir apresenta as curvas de variação de taxa de acerto para diferentes valores de limiares de IoU considerados como acerto de rastreamento. O eixo x representa os diferentes valores de IoU, e o eixo y representa as taxas de acerto. A linha tracejada representa o rastreador de objetos sem peso, mostrando resultados competitivos com outros rastreadores.

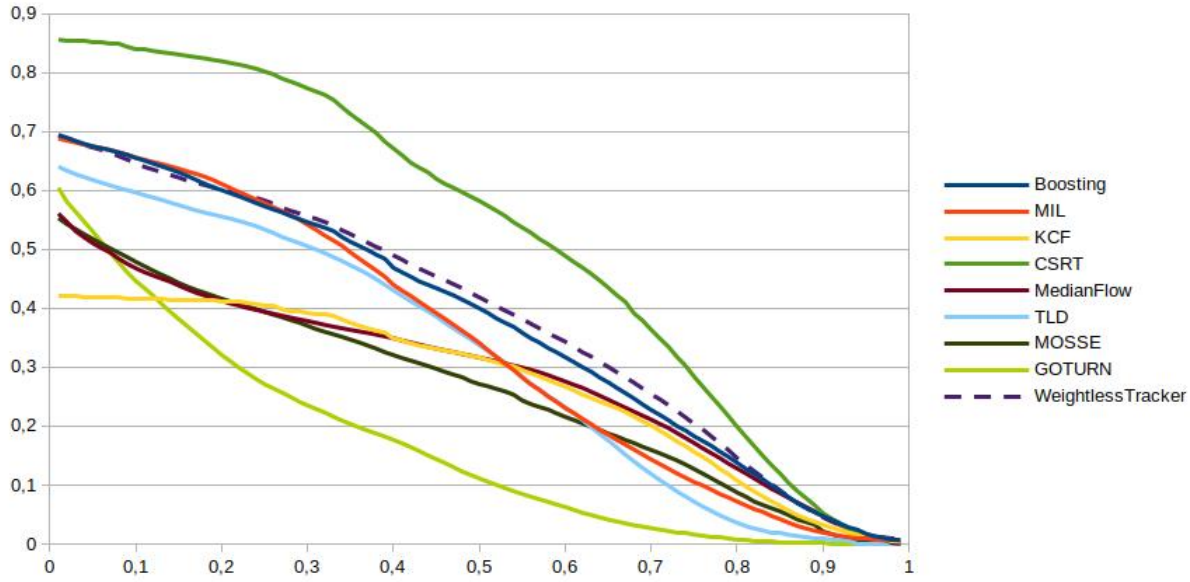

Figura 6.38: Curvas de avaliação dos rastreadores. Cada curva representa um rastreador avaliado nos vídeos do dataset OTB100. Os valores do eixo y representam as taxas de acerto para diferentes valores de IoU no eixo x, considerados como limiares de acertos de rastreamento.

Pode-se observar na figura acima que, quanto maiores os valores de IoU considerados para determinar uma localização de objeto como correta, menores são as taxas de acerto obtidas pelos rastreadores e que a linha tracejada representando o rastreador de objetos sem pesos, mostra que esta abordagem de rastreamento gera resultados competitivos com outros rastreadores em situações envolvendo problemas complexos de rastreamento, como os existentes no dataset OTB100.

Além dos resultados de acurácia através da métrica IoU, o desempenho dos rastreadores também foi medido em número de frames processados por segundo. Todos os testes foram realizados em um notebook Intel Core i7-8750H com 16 GB de memória RAM e os resultados foram os seguintes:

Rastreador	**FPS**
Boosting	66,3139
MIL	33,7172
KCF	176,279
CSRT	61,2326
MedianFlow	1229,89
TLD	32,4299
MOSSE	2513,82
GOTURN	36,3953
WeightlessTracker	60,2907

Tabela 6.5: Performance em frames por segundo

Pode-se observar que o rastreador sem pesos obteve um desempenho competitivo com outros rastreadores, mostrando um bom equilíbrio entre acurácia e performance em frames por segundo. Destaca-se ainda que o grande diferencial do rastreador apresentado nesta tese se dá por conta da possibilidade de obtenção de modelos dos objetos rastreados, sem perdas significativas de desempenho, viabilizando o entendimento sobre as estruturas visuais dos alvos rastreados.

Capítulo 7

Conclusões

Neste capítulo, são apresentados um resumo da tese com as suas principais contribuições, além de perspectivas para possíveis continuações da pesquisa com trabalhos futuros.

7.1 Resumo

Esta tese apresentou um método de criação de modelos visuais descritivos dos aspectos de um objeto rastreado. A principal utilidade dos modelos apresentados são as possibilidades de construção de sistemas computacionais que possuem um entendimento visual sobre as formas dos objetos.

As compreensões sobre os aspectos dos objetos são triviais para serem executadas por seres humanos, porém, de alta complexidade para serem executadas por sistemas de visão computacional. Desta forma, esta tese apresentou um método para proporcionar aos sistemas computacionais, o aprendizado sobre os aspectos dos objetos observados, aprendendo como são as representações visuais dos diferentes aspectos de um mesmo objeto e entendendo as relações entre cada aspecto, possibilitando que ao se apresentar um determinado formato de um objeto, seja possível prever outras formas que podem ser alcançadas por possíveis movimentações ou até mesmo compreender quais as partes dos objetos que estão sofrendo oclusão nos cenários observados. Os experimentos realizados demonstraram que os modelos criados fornecem uma maneira adequada de se representar os objetos observados de maneira que, toda a criação dos modelos foi feita em tempo real, utilizando-se do rastreamento dos objetos em vídeo, sem a utilização de nenhum tipo de treinamento prévio.

7.2 Contribuições

Esta tese contribuiu com a incorporação de novas funcionalidades ao rastreador de objetos baseado nos discriminadores WiSARD [17], desenvolvendo um modo de detectar objetos observando os frames por inteiro em baixa resolução, e acrescentando a possibilidade de detecção dos objetos rastreados em diferentes escalas de tamanho, mantendo as características de rastreamento em tempo real e melhorando a precisão do rastreamento quando ocorrem mudanças de escala nos alvos perseguidos.

Além das funcionalidades de rastreamento utilizando o modelo WiSARD, a principal contribuição apresentada nesta tese foi o desenvolvimento de um método para a criação de representações dos objetos rastreados, baseando-se no algoritmo de clusterização ClusWiSARD, gerando modelos visuais representativos dos objetos observados. Nesta abordagem, as localizações dos alvos determinadas pelo módulo rastreador são passadas para o algoritmo de modelagem, onde os aspectos são determinados e agrupados em discriminadores. Esses discriminadores representantes dos aspectos permitem ainda uma representação visual, utilizando as imagens mentais obtidas pelo modelo DRASiW. Os aspectos determinados pela ClusWiSARD ainda possuem relações entre si através das transições adicionadas aos modelos, possibilitando a obtenção de um entendimento sobre as estruturas visuais apresentadas e permitindo que estes modelos sejam utilizados em aplicações que necessitem das informações aprendidas sobre os aspectos objetos, inclusive com a possibilidade de visualização de partes que estejam sofrendo oclusão.

7.3 Trabalhos futuros

Dentre as possibilidades de trabalhos futuros, estão o desenvolvimento de pesquisas para avaliar a aplicabilidade dos modelos em outros sistemas como por exemplo, aplicações de robótica envolvendo a manipulação de objetos ou até mesmo aplicações que contenham uma base de dados de modelos de objetos para realizar buscas em vídeo. Dentro desse contexto surge a necessidade de explorar o problema de transferência de conhecimento [64, 65] onde os objetos são modelados em um determinado cenário e são utilizados para detecção e rastreamento em diferentes ambientes. O problema de transferência do conhecimento também deve ser avaliado, quando os modelos forem utilizados em um mesmo ambiente, porém, sendo visualizados por múltiplas câmeras posicionadas em diferentes posições. Dessa forma, as novas pesquisas devem contemplar formas de trocas de informação entre as câmeras de maneira que seja possível por exemplo, rastrear um alvo passando por grandes cenários, monitorados por múltiplas câmeras.

Possíveis trabalhos futuros envolvem também o desenvolvimento de outros mé-

todos para a construção dos modelos, como por exemplo, a utilização de múltiplas câmeras posicionadas em diferentes pontos de observação de um mesmo objeto para construir os modelos de forma colaborativa, utilizando simultaneamente as informações provenientes de diferentes câmeras. Outra possibilidade de modificação na construção dos modelos envolve a utilização de sensores como câmeras RGB-D, que permitem a obtenção de informações de profundidade dos objetos rastreados. Essas informações de profundidade podem gerar modelos de imagens mentais mais realistas, utilizando diferentes tons de cinza para representar os pixels mais próximos ou afastados do ponto de captura de imagens. A construção de modelos de imagens mentais mais completos gera expectativas de que os sistemas utilizadores desses modelos consigam realizar as tarefas de visão computacional com boa confiabilidade.

Dentro do contexto de visualização de partes escondidas de um objeto, onde esta tese utilizou a divisão de um alvo em subdiscriminadores, surge a possibilidade de explorar outras formas de divisão de um objeto rastreado, como por exemplo, através de Voronoi Tessellation [66] ou com variações no tamanho da grade de subdiscriminadores.

Partindo-se do estado atual do trabalho, outras pesquisas de menor complexidade de desenvolvimento em relação às citadas anteriormente ainda envolvem a utilização de múltiplos modelos de objetos em uma mesma cena, identificando e visualizando as oclusões sofridas por cada objeto; a utilização de datasets de imagens de objetos para realização de treinamento offline para realizar a construção dos modelos antes da sua utilização em tempo real; e a implementação de versões do sistema para GPUs [67], a fim de obter melhores performances na criação e utilização dos modelos de objetos.

7.4 Considerações finais

Os discriminadores do modelo WiSARD se mostraram bastante adequados para serem utilizados em aplicações de aprendizado de máquina que necessitem de um processamento em tempo real. A metodologia de criação dos modelos a partir das imagens mentais resultou em adequadas representações para os aspectos de objetos nunca vistos anteriormente, possibilitando um entendimento sobre as estruturas visuais de cada objeto observado. Os modelos mentais apresentados nesta tese fornecem uma nova forma de se representar objetos do mundo real para serem utilizados em sistemas computacionais, sendo esses modelos obtidos através do rastreamento em tempo real. Esta funcionalidade de geração de modelos visuais representativos dos objetos, é o grande diferencial do sistema apresentado em relação a outros rastreadores de objetos, e a incorporação da possibilidade de realizar a modelagem em tempo real a partir do rastreamento em vídeo obteve resultados exitosos e pôde ser

realizada sem perdas significativas de desempenho, gerando expectativas de que a abordagem possa auxiliar no desenvolvimento de novas aplicações onde faz-se necessário utilizar conhecimento sobre as estruturas visuais aprendidas. Por fim, o excelente desempenho das redes neurais sem peso, corroboram para atestar a qualidade do modelo de aprendizado através das memórias RAM, resultando em um cenário com boas perspectivas de avanços nesta área.

Referências Bibliográficas

[1] LIU, H., WANG, L. "Gesture recognition for human-robot collaboration: A review", *International Journal of Industrial Ergonomics*, v. 68, pp. 355–367, 2018.

[2] JANAI, J., GÜNEY, F., BEHL, A., et al. "Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art", *Foundations and Trends® in Computer Graphics and Vision*, v. 12, n. 1–3, pp. 1–308, jul. 2020. ISSN: 1572-2740, 1572-2759. doi: 10.1561/0600000079. Disponível em: <https://www.nowpublishers.com/article/Details/CGV-079>.

[3] RAZA, A., ALLAOUA CHELLOUG, S., HAMAD ALATIYYAH, M., et al. "Multiple Pedestrian Detection and Tracking in Night Vision Surveillance Systems", *Computers, Materials & Continua*, v. 75, n. 2, pp. 3275–3289, 2023. ISSN: 1546-2226. doi: 10.32604/cmc.2023.029719. Disponível em: <https://www.techscience.com/cmc/v75n2/52020>.

[4] CHEN, T., XU, J., AGRAWAL, P. "A system for general in-hand object re-orientation". In: *Conference on Robot Learning*, pp. 297–307. PMLR, 2022.

[5] NAGABANDI, A., KONOLIGE, K., LEVINE, S., et al. "Deep dynamics models for learning dexterous manipulation". In: *Conference on Robot Learning*, pp. 1101–1112. PMLR, 2020.

[6] XIANG, Y., CHOI, W., LIN, Y., et al. "Data-driven 3d voxel patterns for object category recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1903–1911, 2015.

[7] LIU, F., LIU, X. "Voxel-based 3D detection and reconstruction of multiple objects from a single image", *Advances in Neural Information Processing Systems*, v. 34, pp. 2413–2426, 2021.

[8] YANG, B., LUO, W., URTASUN, R. "Pixor: Real-time 3d object detection from point clouds". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018.

[9] LIU, Y., FAN, B., XIANG, S., et al. "Relation-Shape Convolutional Neural Network for Point Cloud Analysis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] QI, S., NING, X., YANG, G., et al. "Review of multi-view 3D object recognition methods based on deep learning", *Displays*, v. 69, pp. 102053, 2021.

[11] SU, H., MAJI, S., KALOGERAKIS, E., et al. "Multi-View Convolutional Neural Networks for 3D Shape Recognition". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] KOLMOGOROV, V., ZABIH, R. "Multi-camera scene reconstruction via graph cuts". In: *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part III 7*, pp. 82–96. Springer, 2002.

[13] ESTEBAN, C. H., SCHMITT, F. "Multi-stereo 3d object reconstruction". In: *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*, pp. 159–166. IEEE, 2002.

[14] ZOLLHÖFER, M., STOTKO, P., GÖRLITZ, A., et al. "State of the art on 3D reconstruction with RGB-D cameras". In: *Computer graphics forum*, v. 37, pp. 625–652. Wiley Online Library, 2018.

[15] YANG, J., LI, Z., YAN, S., et al. "RGBD Object Tracking: An In-depth Review", 2022. doi: 10.48550/ARXIV.2203.14134. Disponível em: <`https://arxiv.org/abs/2203.14134`>.

[16] DE CARVALHO, R. L., CARVALHO, D. S., MORA-CAMINO, F. A. C., et al. "Online tracking of multiple objects using WiSARD". In: *ESANN 2014, 22st European Symposium on Artificial Neural Networks, Computational Intelligence And Machine Learning*, pp. pp–541, 2014.

[17] NASCIMENTO, D. N. "Um Rastreador Visual Baseado em Redes Neurais sem Peso e Memórias de Prazo", *Dissertação de M.Sc., PESC/COPPE/UFRJ*, 2015.

[18] NASCIMENTO, D. N., DE CARVALHO, R. L., MORA-CAMINO, F., et al. "A WiSARD-based multi-term memory framework for online tracking of objects", *Proceedings: ESANN 2015*, p. 19, 2015.

[19] DE GREGORIO, M., GIORDANO, M. "Wisardrp for change detection in video sequences". In: *25th European symposium on artificial neural networks, computational intelligence and machine learning*, pp. 453–458, 2017.

[20] ANDRADE, M. B. "Sistema de rastreamento visual de objetos baseado em movimentos oculares sacádicos", *Tese de D.Sc., PPGI/UFES*, 2015.

[21] ALEKSANDER, I. "From WISARD to MAGNUS: A family of weightless virtual neural machines". In: *Ram-Based Neural Networks*, pp. 18–30. doi: 10.1142/9789812816849_0002. Disponível em: <https://www.worldscientific.com/doi/abs/10.1142/9789812816849_0002>.

[22] KALAL, Z., MIKOLAJCZYK, K., MATAS, J. "Forward-backward error: Automatic detection of tracking failures". In: *2010 20th international conference on pattern recognition*, pp. 2756–2759. IEEE, 2010.

[23] BOUGUET, J.-Y. "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm", *Intel Corporation*, v. 5, n. 1-10, pp. 4, 2001.

[24] KALAL, Z., MIKOLAJCZYK, K., MATAS, J. "Tracking-Learning-Detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 34, n. 7, pp. 1409–1422, jul. 2012. ISSN: 0162-8828. doi: 10.1109/TPAMI.2011.239. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2011.239>.

[25] BABENKO, B., YANG, M.-H., BELONGIE, S. "Visual tracking with online multiple instance learning". In: *2009 IEEE Conference on computer vision and Pattern Recognition*, pp. 983–990. IEEE, 2009.

[26] BABENKO, B., YANG, M.-H., BELONGIE, S. "Robust object tracking with online multiple instance learning", *IEEE transactions on pattern analysis and machine intelligence*, v. 33, n. 8, pp. 1619–1632, 2011.

[27] GRABNER, H., GRABNER, M., BISCHOF, H. "Real-time tracking via on-line boosting." In: *Bmvc*, v. 1, p. 6, 2006.

[28] HU, W., HU, W., MAYBANK, S. "Adaboost-based algorithm for network intrusion detection", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 38, n. 2, pp. 577–583, 2008.

[29] HENRIQUES, J. F., CASEIRO, R., MARTINS, P., et al. "High-speed tracking with kernelized correlation filters", *IEEE transactions on pattern analysis and machine intelligence*, v. 37, n. 3, pp. 583–596, 2014.

[30] BOLME, D. S., BEVERIDGE, J. R., DRAPER, B. A., et al. "Visual object tracking using adaptive correlation filters". In: *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 2544–2550. IEEE, 2010.

[31] HELD, D., THRUN, S., SAVARESE, S. "Learning to track at 100 fps with deep regression networks". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 749–765. Springer, 2016.

[32] ALAN, L., VOJÍŘ, T., ČEHOVIN, L., et al. "Discriminative correlation filter tracker with channel and spatial reliability", *International Journal of Computer Vision*, v. 126, n. 7, pp. 671–688, 2018.

[33] DO NASCIMENTO, D. N., FRANÇA, F. M. G. "Object modeling through weightless tracking", *Neural Computing and Applications*, mar. 2024. ISSN: 0941-0643, 1433-3058. doi: 10.1007/s00521-024-09601-5. Disponível em: <https://link.springer.com/10.1007/s00521-024-09601-5>.

[34] MCCULLOCH, W. S., PITTS, W. "A logical calculus of the ideas immanent in nervous activity", *The Bulletin of Mathematical Biophysics*, v. 5, n. 4, pp. 115–133, dez. 1943. ISSN: 0007-4985, 1522-9602. doi: 10.1007/BF02478259. Disponível em: <http://link.springer.com/10.1007/BF02478259>.

[35] HORNIK, K., STINCHCOMBE, M., WHITE, H. "Multilayer feedforward networks are universal approximators", *Neural Networks*, v. 2, n. 5, pp. 359–366, jan. 1989. ISSN: 08936080. doi: 10.1016/0893-6080(89)90020-8. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/0893608089900208>.

[36] ALEKSANDER, I., DE GREGORIO, M., FRANÇA, F. M. G., et al. "A brief introduction to weightless neural systems." In: *ESANN*, pp. 299–305, 2009.

[37] CARNEIRO, H. C., FRANÇA, F. M., LIMA, P. M. "Multilingual part-of-speech tagging with weightless neural networks", *Neural Networks*, v. 66, pp. 11–21, jun. 2015. ISSN: 08936080. doi: 10.1016/j.neunet.2015.02.012. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0893608015000465>.

[38] DE SOUZA, D. F. P., FRANCA, F. M., LIMA, P. M. "Real-Time Music Tracking Based on a Weightless Neural Network". In: *2015 Ninth Inter-*

national Conference on Complex, Intelligent, and Software Intensive Systems, pp. 64–69, Blumenau, jul. 2015. IEEE. ISBN: 9781479988709. doi: 10.1109/CISIS.2015.84. Disponível em: <https://ieeexplore.ieee.org/document/7185167/>.

[39] DE AGUIAR, K., FRANCA, F. M. G., BARBOSA, V. C., et al. "Early detection of epilepsy seizures based on a weightless neural network". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4470–4474, Milan, ago. 2015. IEEE. ISBN: 9781424492718. doi: 10.1109/EMBC.2015.7319387. Disponível em: <https://ieeexplore.ieee.org/document/7319387/>.

[40] BARBOSA, R., CARDOSO, D. O., CARVALHO, D., et al. "Weightless neurosymbolic GPS trajectory classification", *Neurocomputing*, v. 298, pp. 100–108, jul. 2018. ISSN: 09252312. doi: 10.1016/j.neucom.2017.11.075. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0925231218302194>.

[41] VILLON, L. A., SUSSKIND, Z., BACELLAR, A. T., et al. "A conditional branch predictor based on weightless neural networks", *Neurocomputing*, v. 555, pp. 126637, 2023. ISSN: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.126637. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231223007609>.

[42] MIRANDA, I. D., ARORA, A., SUSSKIND, Z., et al. "LogicWiSARD: Memoryless Synthesis of Weightless Neural Networks". In: *2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pp. 19–26, 2022. doi: 10.1109/ASAP54787.2022.00014.

[43] BACELLAR, A., SUSSKIND, Z., VILLON, L., et al. "Distributive Thermometer: A New Unary Encoding for Weightless Neural Networks". pp. 31–36, 01 2022. doi: 10.14428/esann/2022.ES2022-94.

[44] SUSSKIND, Z., BACELLAR, A. T., ARORA, A., et al. "Pruning weightless neural networks", *ESANN 2022 proceedings*, 2022.

[45] SUSSKIND, Z., ARORA, A., MIRANDA, I. D. S., et al. "Weightless Neural Networks for Efficient Edge Inference". In: *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques*, PACT '22, p. 279–290, New York, NY, USA, 2023. Association for Computing Machinery. ISBN: 9781450398688. doi: 10.1145/3559009.3569680. Disponível em: <https://doi.org/10.1145/3559009.3569680>.

[46] ALEKSANDER, I., M. H. "An introduction to Neural Computing. Second edition ed. Berkshire House, London, UK, Thomson Computer Press". 1995.

[47] WICKERT, I., FRANÇA, F. M. G. "AUTOWISARD: Unsupervised Modes for the WISARD". In: Mira, J., Prieto, A. (Eds.), *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence*, pp. 435–441, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN: 978-3-540-45720-6.

[48] CARDOSO, D. O., CARVALHO, D. S., ALVES, D. S., et al. "Financial credit analysis via a clustering weightless neural classifier", *Neurocomputing*, v. 183, pp. 70–78, 2016. ISSN: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.06.105. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231215020421>. Weightless Neural Systems.

[49] CARDOSO, D. O., CARVALHO, D. S., ALVES, D. S., et al. "Credit analysis with a clustering RAM-based neural classifier." In: *ESANN*, 2014.

[50] BGRIECO, B., LIMA, P., GREGORIO, M., et al. "Producing pattern examples from "mental" images", *Neurocomputing*, v. 73, pp. 1057–1064, 2010.

[51] "A probabilistic logic neuron network for associative learning". In: *Neural Computing Architectures: The Design of Brain-Like Machines*, pp. 156–171, 2003.

[52] BOWMAKER, R., COGHILI, G. "Improved recognition capabilities for goal seeking neuron", *Electronics Letters*, v. 28, n. 3, pp. 220–221, 1992.

[53] ALEKSANDER, I. "Ideal neurons for neural computers", *Parallel Processing in Neural Systems and Computers*, pp. 225–228, 1990.

[54] ALEKSANDER, I., MORTON, H. "General neural unit: retrieval performance", *Electronics letters*, v. 19, n. 27, pp. 1776–1778, 1991.

[55] KANERVA, P. *Sparse distributed memory*. MIT press, 1988.

[56] IZQUIERDO, I. A., MYSKIW, J. C., BENETTI, F., et al. "Memória-Tipos e mecanismos-Achados recentes", *Revista USP*, 2013.

[57] "Object Modeling Through Weightless Tracking Dataset". 2023. Disponível em: <https://figshare.com/articles/figure/Object_Modeling_Through_Weightless_Tracking_Dataset/24034317/1>.

[58] WU, Y., LIM, J., YANG, M.-H. "Object Tracking Benchmark", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 37, n. 9, pp. 1834–1848, 2015. doi: 10.1109/TPAMI.2014.2388226.

[59] EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., et al. "The Pascal Visual Object Classes (VOC) Challenge", *International Journal of Computer Vision*, v. 88, n. 2, pp. 303–338, jun. 2010. ISSN: 0920-5691, 1573-1405. doi: 10.1007/s11263-009-0275-4. Disponível em: <http://link.springer.com/10.1007/s11263-009-0275-4>.

[60] GARRETT, C. R., CHITNIS, R., HOLLADAY, R., et al. "Integrated Task and Motion Planning", *Annual Review of Control, Robotics, and Autonomous Systems*, v. 4, n. 1, pp. 265–293, maio 2021. ISSN: 2573-5144, 2573-5144. doi: 10.1146/annurev-control-091420-084139. Disponível em: <https://www.annualreviews.org/doi/10.1146/annurev-control-091420-084139>.

[61] BRADSKI, G. "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2000.

[62] BABENKO, B., YANG, M.-H., BELONGIE, S. "Robust object tracking with online multiple instance learning", *IEEE transactions on pattern analysis and machine intelligence*, v. 33, n. 8, pp. 1619–1632, 2010.

[63] KALAL, Z., MIKOLAJCZYK, K., MATAS, J. "Tracking-learning-detection", *IEEE transactions on pattern analysis and machine intelligence*, v. 34, n. 7, pp. 1409–1422, 2011.

[64] WEISS, K., KHOSHGOFTAAR, T. M., WANG, D. "A survey of transfer learning", *Journal of Big Data*, v. 3, n. 1, pp. 9, dez. 2016. ISSN: 2196-1115. doi: 10.1186/s40537-016-0043-6. Disponível em: <http://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0043-6>.

[65] ZHUANG, F., QI, Z., DUAN, K., et al. "A Comprehensive Survey on Transfer Learning", *Proceedings of the IEEE*, v. 109, n. 1, pp. 43–76, jan. 2021. ISSN: 0018-9219, 1558-2256. doi: 10.1109/JPROC.2020.3004555. Disponível em: <https://ieeexplore.ieee.org/document/9134370/>.

[66] DU, Q., FABER, V., GUNZBURGER, M. "Centroidal Voronoi tessellations: Applications and algorithms", *SIAM review*, v. 41, n. 4, pp. 637–676, 1999.

[67] SANDERS, J., KANDROT, E. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.