

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**UM ALGORITMO PARA O
RASTREAMENTO EM CARDUMES
ATRAVÉS DA ANÁLISE DE IMAGENS
DIGITAIS**

GREGORY DE OLIVEIRA FEIJÓ

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Ciência da
Computação na Pontifícia Universidade Católica do
Rio Grande do Sul.

Orientador: Prof. Dr. Márcio Sarroglia Pinho
Coorientador: Prof. Dr. Isaac Newton da Silva

Porto Alegre
2016

Dados Internacionais de Catalogação na Publicação (CIP)

F297a Feijó, Gregory de Oliveira

Um algoritmo para o rastreamento em cardumes através da análise de imagens digitais / Gregory de Oliveira Feijó. – 2016.
97 p.

Dissertação (Mestrado) – Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Márcio Sarroglia Pinho
Coorientador: Prof. Dr. Isaac Newton da Silva

1. Processamento de Imagens. 2. Algoritmos. 3. Peixes - Comportamento. 4. Informática. I. Pinho, Márcio Sarroglia. II. Silva, Isaac Newton da. III. Título.

CDD 23 ed. 006.6

Loiva Duarte Novak CRB 10/2079
Setor de Tratamento da Informação da BC-PUCRS



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "Um Algoritmo para o Rastreamento em Cardumes Através da Análise de Imagens Digitais" apresentada por Gregory de Oliveira Feijó como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, aprovada em 11/03/2015 pela Comissão Examinadora:

Prof. Dr. Márcio Sarroglia Pinho -
Orientador

PPGCC/PUCRS

Prof. Dr. Isacc Newton da Silva -
Coorientador

FENG/PUCRS

Profa. Dra. Soraia Raupp Musse -

PPGCC/PUCRS

Prof. Dr. Jauvane Cavalcante de Oliveira -

LNCC

Homologada em 05/05/2016, conforme Ata No. 09..... pela Comissão Coordenadora.

Prof. Dr. Luiz Gustavo Leão Fernandes
Coordenador

PUCRS

Campus Central
Av. Itália, 6681 - P32 sala 507 CEP: 90619-900
Fone: (51) 3320-3611 - Fax (51) 3320-3621
E-mail: pggcc@pucrs.br
www.pucrs.br/facinf/ocs

DEDICATÓRIA

Dedico este trabalho aos meus pais, à minha namorada e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Dedico também, este trabalho aos meus colegas e amigos do Grupo de Realidade Virtual, pelas alegrias, tristezas e dores compartilhadas. Com vocês, as pausas entre um parágrafo e outro de produção melhora tudo o que tenho produzido na vida.

AGRADECIMENTOS

Muitas pessoas contribuíram para a realização desta dissertação de mestrado e não poderia deixar de manifestar o meu profundo agradecimento a todas elas.

Primeiramente, eu gostaria de agradecer ao meu orientador Prof. Dr. Márcio Sarrógia Pinho, pelo empenho dedicado à elaboração deste trabalho e aos colegas e amigos do GRV (Grupo de Realidade Virtual) pelo ótimo ambiente de trabalho e ótima convivência que me proporcionaram durante os anos que fiz parte deste grupo.

Agradeço aos meus pais pelo amor, incentivo e apoio incondicional que me proporcionaram ao longo do curso de mestrado. À minha namorada que com muita paciência, amor e carinho me acompanhou do início ao fim do percurso.

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro para a realização deste trabalho possibilitando minha dedicação exclusiva ao mesmo.

Aos funcionários, alunos e professores do PPGCC pela convivência, amizade e experiências compartilhadas ao longo do curso de mestrado.

UM ALGORITMO PARA O RASTREAMENTO EM CARDUMES ATRAVÉS DA ANÁLISE DE IMAGENS DIGITAIS

RESUMO

O monitoramento de peixes em aquários, tem sido utilizado em áreas de pesquisa biológica por exemplo, para entender o efeito do uso de drogas. Este monitoramento requer a obtenção da trajetória destes animais para posterior análise. A utilização de observadores humanos, ainda é o principal método utilizado para este monitoramento. Entretanto, esta não é uma abordagem confiável devido à dificuldade em se manter o foco de atenção de um humano sobre uma fonte de informação por muito tempo. Por esta razão, tem se tornado comum o uso de técnicas de processamento de imagens para a tarefa de monitoramento. O rastreamento de um único peixe, é uma tarefa relativamente simples, que pode ser resolvida com o uso de métodos tradicionais de processamento de imagens e rastreamento. O rastreamento de vários peixes, entretanto, é uma tarefa mais desafiadora. O maior problema, nestes casos, é a manutenção da identidade de cada indivíduo devido às frequentes situações de sobreposição(occlusão) que ocorrem durante o movimento dos animais no aquário. Algumas das abordagens baseiam-se em informações tridimensionais obtidas de múltiplas câmeras, o que requer uma etapa de calibração trabalhosa. Já abordagens com uma única câmera têm dificuldade de tratar corretamente os casos de oclusão, resultando na troca de identidades com frequência. Este trabalho apresenta um método para rastrear um grupo de peixes em um aquário. O método é capaz, manter a identidade correta de cada indivíduo mesmo em casos de oclusão parcial e total. Para manter a identidade de um peixe ao longo do tempo, utiliza-se o Filtro de Kalman que permite estimar qual a posição futura de um objeto com base nas posições anteriores. . Em casos de oclusão, quando houver mais de um peixe em um mesmo local, executa-se um algoritmo de particionamento de regiões, com o objetivo de reconstruir a forma original de cada indivíduo. O resultado do algoritmo foi comparado com um *ground truth* obtido manualmente em dois vídeos. Testes mostram que o método proposto é capaz de manter a identidade dos animais, em 98,04% dos casos de oclusão.

Palavras-Chave: Rastreamento de Múltiplos Objetos, Cardumes, Comportamento de Cardumes, Processamento Digital de Imagens, Tratamento de Oclusão

AN ALGORITHM TO TRACK A GROUP OF SHOALING FISH BY MEANS OF DIGITAL IMAGE ANALYSIS

ABSTRACT

Fish monitoring has been recently used in many biological research fields to understand the effects of drug usage, for example. Monitoring tasks require the full trajectory of these animals for later evaluation. Evaluation by human observers is the main approach used nowadays. However, this is not a reliable approach because humans can not maintain focus on a source of information for too long. For this reason, digital image processing techniques have become a popular approach for monitoring tasks. The tracking of a single fish is a relatively simple problem that may be solved with traditional image processing techniques. On the other hand, the tracking of a group of fish is much more challenging. The biggest problem is to maintain each individual's identity due to the frequent overlapping (occlusion) situation that occurs while these animals move inside the tank. Some known approaches use three-dimensional information obtained by multiple cameras which requires a laborious camera calibration step. Other approaches based on a single camera, can not correctly handle occlusion, resulting in a frequent identity swap between fish. This work presents a multi-object tracking method to track a group of fish in a tank. The proposed method is capable of maintaining the correct identity of each fish even in partial and full occlusion situations. In order to keep the correct identity, we take advantage of the Kalman Filter by estimating the future position of each fish based on its previous one. When there are more than one fish in the same region in the frame image, a partitioning algorithm is responsible for re-establishing each animal's pose. The proposed algorithm was compared against a manually labeled ground truth in two videos. Preliminary tests show that the proposed method is capable of maintaining the animals identity in 98,04% of the occlusion cases.

Keywords: Multi-Object Tracking, Shoaling Fish, Shoaling Behavior, Digital Image Processing, Image Processing, Occlusion Handling

LISTA DE FIGURAS

Figura 1 - Exemplo de oclusão entre peixes.....	26
Figura 2 - O algoritmo de busca utilizado por Kato et al. [16].	31
Figura 3 - Ilustração do modelo de estimativa de trajetória desenvolvido por Delcourt. T0, T1 e T2 representam os instantes de tempo 0, 1 e 2 respectivamente, A e B representam dois peixes distintos.....	32
Figura 4 - Algoritmo para determinar direção do movimento de peixes.	35
Figura 5 - Ilustração do aparato desenvolvido para o monitoramento dos animais com espelhos	36
Figura 6 - Exemplo de Imagens provenientes das câmeras virtuais e da câmera real. (a) câmera virtual superior, (b) câmera virtual lateral, (c) câmera real.....	37
Figura 7 - Arquitetura para o rastreamento de objetos	40
Figura 8 - Exemplos de representação de objetos através de formas geométricas primitivas (a) e (b) e contornos (c).	41
Figura 9 - Representação de objetos através de blobs. (a) imagem de um cenário. (b) objetos de interesse segmentados.	41
Figura 10 - Representação de objetos através de templates.....	42
Figura 11 - Representação de objetos através de histogramas.	42
Figura 12 - Imagens retiradas do primeiro cenário	49
Figura 13 - Imagens retiradas do segundo cenário	50
Figura 14 - Conjunto de dados obtidos através do método CAMShift para rastrear um único peixe.	51
Figura 15 - Ilustração da janela de busca divergindo quando outros animais se aproximam.	52
Figura 16 - Imagens capturadas de peixes em ambos os cenários. (a) cenário com um peixe, (b) cenário com múltiplos peixes	52

Figura 17 - Conjunto de imagens obtidas através do método SURF para o rastreamento de um peixe.	53
Figura 18 - Comparação entre blobs obtidos por duas técnicas distintas.....	54
Figura 19 - Conjunto de dados obtidos através do filtro de Kalman para rastrear um único peixe.....	55
Figura 20 - Conjunto de imagens obtidas através do filtro de Kalman para rastrear múltiplos peixes	56
Figura 21 - Conjunto de imagens obtidas após adaptação do Filtro de Kalman para o rastreamento de múltiplos animais.	57
Figura 22 - Exemplo de blobs em que há oclusão ou sobreposição entre peixes no aquário.	58
Figura 23 - Diagrama do método proposto.	61
Figura 24 - Exemplo de peixes detectados na etapa 1 do algoritmo.	61
Figura 25 - Exemplo de situação em que há mais instâncias do Filtro de Kalman que blobs-candidatos.	62
Figura 26 - Modelo de fundo capturado no início do processo de rastreamento.	62
Figura 27 - Resultado da técnica de remoção de fundo.	63
Figura 28 - Exemplo de blobs ruidosos devido a reflexão da luz.....	63
Figura 29 - Exemplo de blobs pequenos	64
Figura 30 - Comparação entre Centróide e Mediana Geométrica	64
Figura 31 - Mediana Geométrica e Orientação de um Blob.....	65
Figura 32 - Processo de translação do blob obtido no <i>frame</i> anterior.....	66
Figura 33 - Processo de seleção de blobs realizado através do método de intersecção. .	67
Figura 34 - Processo de seleção de blobs com base no blob mais próximo.	68
Figura 35 - Exemplo de dois Filtros de Kalman apontando para um o mesmo blob-candidato.	69
Figura 36 - Exemplo de expansão de um blob de um peixe-zebra com diferentes elementos estruturantes.....	70

Figura 37 - Ciclo trigonométrico utilizado para definir o elemento estruturante de um blob.	71
Figura 38 - Exemplos do método de seleção de máscaras de expansão.....	72
Figura 39 - Passo a passo do processo de expansão de um blob.	74
Figura 40 - Sequência de iterações realizadas por um blob até atingir sua área histórica.	75
Figura 41 - Exemplo de resultado obtido após a realização do processo de expansão em dois blobs.	76
Figura 42 - Ilustração do cenário onde foram obtidos os vídeos.	79
Figura 43 - Câmera iLook 1321 V2 utilizada para capturar os vídeos.	79
Figura 44 - Exemplo de atribuição falha com o método MG e correta com o método INT.	83
Figura 45 - Exemplo de atribuição falha com o método INT e correta com o método MG.	84
Figura 46 - Exemplo de particionamento em que um blob expande na direção errada.	86
Figura 47 - Exemplo de resultados do particionamento de blobs com dois peixes.	87
Figura 48 - Exemplo de resultados do particionamento de blobs com três peixes.	88

LISTA DE TABELAS

Tabela 1 - Conjunto de métricas avaliadas em experimentos com cardumes.	30
Tabela 2 - Conjunto de informações que compõem o filtro de Kalman para Rastreamento	46
Tabela 3 - Condições para expansão do pixel	73
Tabela 4 - Informações sobre os vídeos A e B testados.	80
Tabela 5 - Quantidade total de erros de cada método em comparação com o <i>ground truth</i>	81
Tabela 6 - Performance do método proposto em comparação com os métodos de seleção de candidatos avaliados.	82
Tabela 7 - Resultados obtidos pelo método proposto em comparação com o <i>ground truth</i>	85

LISTA DE SIGLAS

C - Centroide

HSV - Hue, Saturation e Value

INT - Intersecção

LSD - Dietilamida do ácido lisérgico

MG - Mediana Geométrica

QA - Quantidade de Acertos

QE - Quantidade de Erros

SUMÁRIO

1	INTRODUÇÃO.....	25
1.1	ORGANIZAÇÃO DO TEXTO	27
2	MONITORAMENTO DE CARDUMES	29
2.1	EXPERIMENTOS REALIZADOS COM CARDUMES	29
2.2	MONITORAMENTO AUTOMÁTICO DE CARDUMES BASEADO EM VÍDEO	30
2.2.1	<i>Técnicas 2D</i>	31
2.2.2	<i>Técnicas 3D</i>	34
3	RASTREAMENTO DE OBJETOS EM VÍDEO.....	39
3.1	MODELAGEM DO OBJETO.....	40
3.1.1	<i>Representação</i>	40
3.1.2	<i>Propriedades</i>	42
3.2	DETECÇÃO DO OBJETO	43
3.3	ROTULAGEM DO OBJETO	44
4	AVALIAÇÃO DE MÉTODOS DE RASTREAMENTO APLICADOS A PEIXES.....	49
4.1.1	<i>CAMShift</i>	50
4.1.2	<i>Speed-Up Robust Features</i>	52
4.1.3	<i>Filtro de Kalman</i>	53
4.1.4	<i>Análise dos Resultados</i>	57
5	MÉTODO DESENVOLVIDO.....	59
5.1	RESUMO DO MÉTODO.....	60
5.2	DETECÇÃO DE BLOBS E EXTRAÇÃO DE CARACTERÍSTICAS	62
5.3	SELEÇÃO DO MELHOR BLOB-CANDIDATO	65
5.4	PARTICIONAMENTO DE BLOB.....	68
5.5	PÓS-PROCESSAMENTO	77

6	RESULTADOS	79
6.1	AVALIAÇÃO DO MÉTODO DE SELEÇÃO DE CANDIDATOS	80
6.2	AVALIAÇÃO DO MÉTODO DE PARTICIONAMENTO	84
7	CONCLUSÕES E TRABALHOS FUTUROS	89
	REFERÊNCIAS BIBLIOGRÁFICAS	91

1 INTRODUÇÃO

Diversas áreas do conhecimento têm se beneficiado com os recentes avanços nas tecnologias de *hardware* e *software* observadas nos últimos anos. Entre estas, está a Biologia Comportamental, na qual são realizados estudos para o reconhecimento de padrões no comportamento de animais [1].

Segundo o Comitê para o Uso de Animais de laboratório em Pesquisas Biomédicas e Comportamentais, do inglês *CULABBR* [2], experimentos conduzidos com animais de laboratório são muito comuns e têm como principal objetivo melhorar a saúde de seres humanos e outros animais, além de permitir a aquisição de conhecimentos que não seriam possíveis de outro modo. Geralmente, este tipo de estudo é realizado através de monitoramento conduzido por um observador humano [3] [4] [5]. A utilização de observadores humanos, entretanto, não é uma abordagem muito confiável devido à dificuldade destes em manter o foco de atenção sobre uma fonte de informação por mais de alguns segundos [6] [7]. Em função disto, têm surgido, nos últimos anos, uma série de ferramentas, baseadas em processamento de imagens, para servirem como apoio à realização do monitoramento de animais [8] [9]. Estas ferramentas são, via de regra, menos suscetível a erros se comparadas com observadores humanos. Dentre os animais usados em experimentos de monitoramento, estão os peixes. Nestes experimentos, são monitoradas diversas propriedades como velocidade, direção e periodicidade do nado de peixes, com o intuito de avaliar, por exemplo, a eficácia de novas drogas [10], os níveis de oxigênio dissolvido na água [11] ou de aprender sobre as características apresentadas por esses animais quando estão em grupos [12].

Além de monitorar o comportamento individual de peixes, é comum a análise do comportamento destes animais em cardumes [13] [14]. Esta análise pode ser realizada manualmente [15] ou através de um processo parcialmente automatizado [9], em que o pesquisador identifica a posição dos animais em cada imagem e uma ferramenta realiza a quantificação das métricas posteriormente.

O rastreamento de vários peixes é uma tarefa mais desafiadora se comparada ao rastreamento de um único animal. O maior problema, nestes casos, é a manutenção da identidade de cada indivíduo devido às frequentes situações de oclusão entre os indivíduos. Neste contexto, entende-se por oclusão o fato de um ou mais indivíduos se cruzarem ou

ficarem muito próximos uns dos outros, fazendo com que a imagem destes não permita sua individualização, como se pode observar no detalhe da Figura 1.

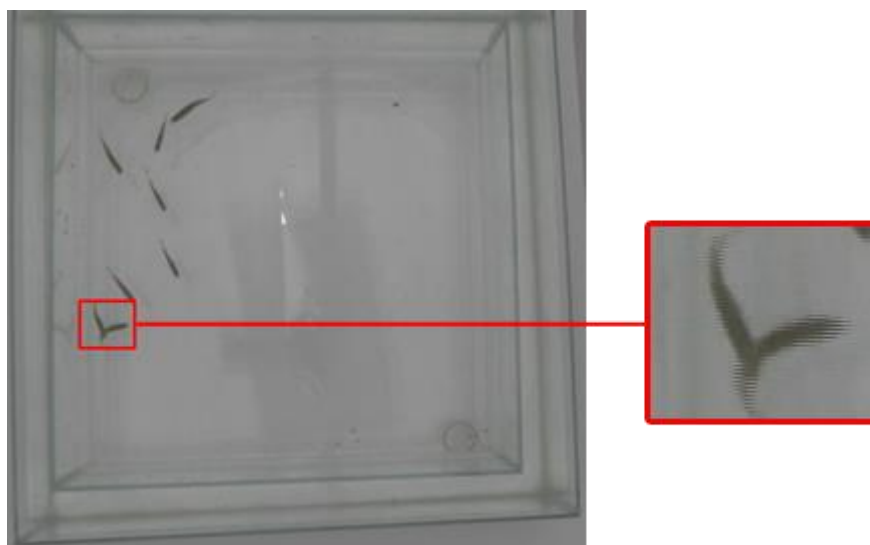


Figura 1 - Exemplo de oclusão entre peixes.

Nas abordagens encontradas na literatura para o rastreamento de diversos peixes, as situações de oclusão são tratadas de diferentes formas. Kato et al. [16], por exemplo, buscam separar os blobs em oclusão através de operações morfológicas de erosão e dilatação, entretanto, o método não funciona quando a sobreposição entre os peixes é quase total. No trabalho de Escudero et al. [17], para cada peixe, um identificador é calculado com o objetivo de diferenciá-lo dos demais. Quando ocorre uma oclusão, o trecho é ignorado. Ao final do processo os diversos trechos identificados para cada peixe são unidos formando a trajetória completa do animal. O método entretanto pode falhar para peixes muito similares em cor, tamanho e textura, resultando em atribuições incorretas de identidade. Alguns autores ainda exploram o uso de duas ou mais imagens do mesmo cenário buscando tratar a oclusão em uma imagem através das outras [18] [19], entretanto, a configuração do cenário pode ser bastante trabalhosa [18]. Em face deste panorama, o presente trabalho desenvolveu um algoritmo, baseado em processamento digital de imagens, capaz de manter a identidade de cada indivíduo de um cardume mesmo em situações de oclusão.

Na abordagem apresentada, a trajetória de um peixe é gravada em um vídeo. A seguir este vídeo é processado removendo-se a imagem do fundo do aquário, o que resulta em sequência de quadros, ou *frames*, com cada um deles contendo um conjunto de pequenas imagens, denominadas *blobs*, que representam os peixes.

Para manter a identidade de um peixe entre um *frame* e o seguinte, utiliza-se um Filtro de Kalman que permite estimar qual dos blobs pertence a cada animal. Apesar de funcionar em muitos casos, os testes realizados em aquários com 10 animais da espécie peixe-zebra, durante 2(dois) minutos, demonstraram que a aplicação simples do Filtro gerou 50 situações de erro. Por isto, criou-se um algoritmo que corrige esta estimativa com base na intersecção da imagem de cada peixe de um *frame* e os blobs existentes no *frame* seguinte. Esta correção permitiu reduzir os erros para apenas 7 ocorrências nos 2(dois) minutos de vídeo.

Naqueles *frames* em que ocorrem casos de oclusão, haverá blobs contendo mais de um peixe, e por consequência, menos blobs do que os peixes que existem no aquário. Nestes casos, um algoritmo de divisão de blobs é executado com o objetivo de reconstruir a forma original de cada indivíduo. Essa reconstrução é realizada iterativamente através de sucessivas operações morfológicas de dilatação utilizando um elemento estruturante que é gerado dinamicamente com base em informações ligadas à forma do animal, obtidas dos *frames* anteriores. A taxa de sucesso deste processo de divisão foi de 98,04%. Estes resultados indicam que o algoritmo proposto é capaz de manter a referência correta de cada blob antes, durante e depois da ocorrência de oclusão.

1.1 Organização do Texto

O conteúdo do texto aqui apresentado é organizado da seguinte maneira. No início do capítulo 2, é apresentada uma introdução sobre o monitoramento de cardumes, destacando sua importância e as principais métricas avaliadas nesses experimentos. No final do capítulo, uma revisão do estado da arte apresenta algumas técnicas de rastreamento em duas e três dimensões baseadas no processamento digital de imagens. No capítulo 3, cada um dos estágios do processo de rastreamento em vídeo é descrito, desde a modelagem de objetos de interesse até a rotulagem desses objetos. Ainda no capítulo 3, são avaliados os métodos de rastreamento tradicionais aplicados no contexto de monitoramento de cardumes. Os capítulos 4 e 5 apresentam o método proposto e os resultados obtidos em dois vídeos distintos, respectivamente. As principais limitações do método proposto são apresentadas na seção 6, enquanto que os trabalhos futuros é apresentado no capítulo 7.

2 MONITORAMENTO DE CARDUMES

Qualquer grupo de peixes que permaneça junto por razões sociais é denominado de cardume. Este é um termo análogo à alcateia ou rebanho, que representam coletivos de lobos e ovelhas respectivamente. Um cardume pode fornecer diversos benefícios para o grupo, como, por exemplo, facilidade na busca por alimento, reprodução e defesa contra predadores [20]. Por conta disso, diversos estudos buscam compreender o comportamento desses animais em grupos.

Na seção 2.1, serão apresentados alguns estudos com peixes e cardumes que têm sido conduzidos nos últimos anos, destacando as principais métricas utilizadas para avaliar o comportamento desses animais. Já na seção 2.2, são apresentados métodos baseados no processamento de imagens, que compõem o estado da arte no rastreamento de peixes e cardumes.

2.1 Experimentos realizados com cardumes

Diversos experimentos com cardumes têm sido conduzidos nos últimos anos, em especial o que diz respeito ao efeito de drogas no comportamento desses animais [10]. Os peixes-zebra, por exemplo, apresentam certa sensibilidade a drogas responsáveis por causar dependência química, como álcool, nicotina e LSD. Os dois primeiros apresentaram uma diminuição significativa na velocidade de um cardume [10], enquanto uma redução aparente na coesão do grupo foi percebida ao expor os animais ao LSD [21]. Outros estudos observaram uma diminuição na tendência desses animais a afastar-se do cardume quando sob o efeito de um desregulador endócrino [15].

Alguns fatores podem influenciar certos peixes na formação de cardumes. Em alguns experimentos, por exemplo, foi observado que o Barrigudinho tende a discriminar cardumes grandes de pequenos e preferir parceiros com tamanho semelhante ao seu [3]. Similarmente ao barrigudinho, o Peixe Mosquito também apresenta uma preferência por cardumes grandes, porém, adicionalmente procura formar grupo com peixes mais sociáveis [22]. Algumas espécies de peixes podem divergir em suas escolhas de acordo com experiências anteriores. O Acará, por exemplo, quando criado apenas com peixes de mesma coloração, apresentará preferência a grupos de mesma cor, ao passo que se criados com peixes de diferente coloração, essa preferência não é manifestada [23].

Uma revisão sistemática realizada para identificar as métricas mais comumente empregadas na avaliação comportamental de cardumes indicou uma grande quantidade de parâmetros. Uma lista completa com os resultados dessa revisão é apresentada na Tabela 1.

Tabela 1 - Conjunto de métricas avaliadas em experimentos com cardumes.

Métricas	Autores
Centro de massa	[24]
Coesão do cardume	[10], [25], [26]
Distância entre membros do cardume	[27], [10], [25], [26], [28], [21]
Distância para os membros próximos	[13], [4], [10], [29], [30]
Latência para cruzar linha	[15]
Latência para cruzar a primeira linha	[31]
Posição da cabeça em relação a marcadores	[14]
Orientação do cardume	[13], [10]
Quantidade de cardumes	[4]
Quantidade de colisões entre peixes	[13]
Quantidade de cruzamentos de linha	[31]
Quantidade de entradas em uma região	[5]
Quantidade de peixes no cardumes	[4]
Quantidade de transições entre regiões	[15]
Raio do cardume	[13]
Tempo gasto em uma região	[15], [3], [23], [31], [22]
Velocidade Média	[10]

2.2 Monitoramento Automático de Cardumes Baseado em Vídeo

Peixes são animais que possuem maior liberdade de locomoção se comparados com roedores, por exemplo, visto que os primeiros podem se mover em três dimensões, enquanto roedores podem se mover em apenas duas. Por conta disso, é comum encontrar tanto técnicas de rastreamento em duas dimensões quanto em três dimensões para quantificar a atividade motora de peixes.

As abordagens que serão apresentadas aqui propõem soluções para quantificar a atividade de cada indivíduo do grupo [32] [33] [18] [34], do grupo como um todo [9] ou identificar episódios particulares [35] [16]. A seguir, serão apresentadas algumas dessas técnicas subdivididas em duas categorias: 2D e 3D.

2.2.1 Técnicas 2D

Em abordagens 2D, as imagens são capturadas ao longo do experimento através de uma única câmera com o intuito de identificar o/os indivíduo/s e inferir a sua posição no espaço.

Em 2004, Kato et al. [16] propuseram um sistema para identificar e quantificar o comportamento de até dois peixes-zebra a partir de uma câmera posicionada sobre o aquário. O sistema é capaz de identificar a posição de cada animal, tratar oclusão e identificar episódios de perseguição entre dois peixes utilizando como base um método de remoção de fundo para separar os peixes do restante do cenário.

Após separar os peixes, a detecção dos animais é realizada através de um algoritmo de busca baseado em saltos, proposto com o intuito de reduzir o tempo de processamento necessário para avaliar toda a imagem. Nesse algoritmo, a imagem é percorrida avaliando-se 1 em cada 4 pixels da imagem (Figura 2A) e para cada pixel p candidato a peixe, avalia-se os 4 vizinhos diagonais (V_D) de p (Figura 2B) e os 4 vizinhos próximos (V_P) de p (Figura 2C) com o intuito de agrupar esses candidatos em um único objeto. A seguir, para cada um dos 4 pontos V_D , são avaliados os seus V_P . O resultado deste processo pode ser visualizado na Figura 2D.

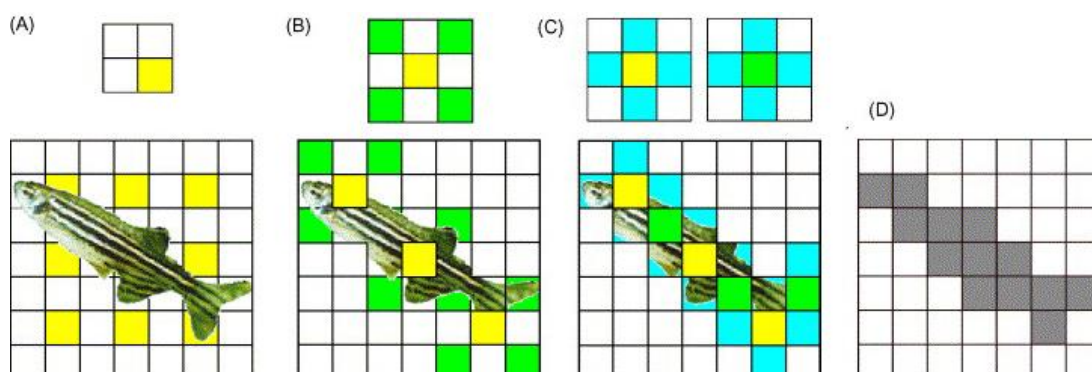


Figura 2 - O algoritmo de busca utilizado por Kato et al. [16].

Fonte: Adaptado de Kato et al. [16]

A técnica proposta pelos autores funciona nos casos em que os peixes estão afastados, porém, quando estes se encontram muito próximos, a técnica resultará em uma região que conterà os dois peixes cujas imagens estarão sobrepostas. Para evitar esse problema, os autores utilizaram um algoritmo baseado em operações morfológicas para separar as imagens dos dois peixes. Basicamente são realizadas operações de erosão até que a região seja dividida em duas regiões distintas. Uma vez identificados os peixes, a posição espacial é definida pelo centro de gravidade de cada peixe identificado. Este algoritmo não funciona caso a região que contém os dois peixes não possa ser separada em duas.

Para detectar os episódios de perseguição o algoritmo monitora a distância entre os peixes, que deve estar entre o intervalo de 1 a 10cm, e o ângulo formado entre os peixes, que deve ser menor que 50° . Ambos os valores foram definidos empiricamente.

O trabalho de Kato et al. só funciona com dois peixes, o que nem sempre é suficiente em um experimento com grupos. Por isto, Delcourt et al. [33] propuseram um sistema que realiza a detecção da trajetória com base na velocidade e na direção do animal nos últimos dois instantes de tempo (Figura 3), gerando, assim, um conjunto de possíveis candidatos. O candidato que estiver mais próximo à posição real é utilizado para inferir o destino final do animal.

O sistema foi desenvolvido para analisar o comportamento de peixes, como o tilápia-do-nylo e é capaz de rastrear até 100 peixes ao longo do experimento. Porém, os autores não tratam a questão de oclusão que se torna mais frequente à medida em que se tem mais peixes no grupo.

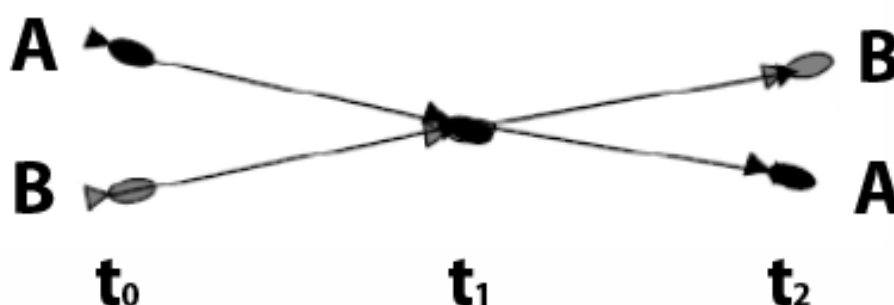


Figura 3 - Ilustração do modelo de estimativa de trajetória desenvolvido por Delcourt. T_0 , T_1 e T_2 representam os instantes de tempo 0, 1 e 2 respectivamente, A e B representam dois peixes distintos.

No que diz respeito à oclusão, uma forma trata-la é utilizar uma abordagem interativa, em que parte do processo é realizado manualmente. Um exemplo disto é o trabalho de Miller et al. [9], utilizado para quantificar parâmetros relacionados à coesão de cardumes de peixes-zebra através de imagens obtidas por uma câmera posicionada sobre o aquário. A identificação de cada peixe é realizada por um observador humano que aponta a posição de cada peixe nas imagens através de um mouse, evitando, assim, que a precisão da ferramenta seja influenciada por ruídos e oclusão de peixes por exemplo. Uma vez identificados os peixes, o sistema calcula a distância de um peixe para cada um dos outros e a distância para os vizinhos próximos.

Também focado no rastreamento de múltiplos indivíduos Chen et al. [34], propuseram uma ferramenta para quantificar o comportamento de peixes com o intuito de monitorar a qualidade da água em grandes tanques. A ferramenta é capaz de extrair informações como, aceleração, ângulo de virada, trajetória e velocidade de cada peixe, utilizadas como indicadores do nível de poluição da água. No trabalho de Chen et al., os peixes são separados do restante do cenário através de uma técnica de remoção de fundo por diferença de *frames* e o rastreamento dos animais é feita através de um filtro de partículas [36] responsável por estimar um conjunto de possíveis candidatos para cada animal.

Uma abordagem diferente, mas também voltada ao rastreamento de múltiplos indivíduos é apresentada por Escudero et al. [17]. Ao invés de tentar identificar os animais durante os momentos de oclusão, estes trechos dos vídeos são ignorados. Nos trechos restantes, o algoritmo busca obter identificadores, ou *fingerprints*, de cada peixe, o que permite unir os trechos pertencentes a cada animal, ao final do processamento. Os identificadores são calculados a partir de um conjunto de imagens referência de cada peixe, que são obtidas nos *frames* iniciais do vídeo. Para cada *frame* subsequente, a imagem é segmentada através de uma técnica de remoção de fundo simples. Para comparar cada indivíduo resultante da segmentação, com as imagens de referência, e assim identificar que peixe está em cada trecho, o método utiliza uma técnica baseada na correlação dos histogramas de cor [37]. Embora o algoritmo apresente bons resultados, este depende fortemente dos identificadores, o que leva à necessidade de se ter imagens com alto grau de detalhamento e um cenário com iluminação controlada.

2.2.2 Técnicas 3D

No caso do rastreamento de objetos em 2D, a trajetória real do animal é projetada em um plano perpendicular ao eixo da câmera, perdendo, assim, a informação que esteja no mesmo eixo da câmera. Dessa forma, é improvável mensurar a quantidade de movimento de forma fidedigna devido à perda de informação resultante desse processo. Técnicas que exploram o espaço tridimensional são muitas vezes empregadas para quantificar o comportamento de forma mais precisa, pois o movimento real de objetos se dá em 3D.

Nessas abordagens, geralmente são utilizadas duas ou mais câmeras posicionadas ao redor do cenário com o intuito de capturar imagens 2D em ângulos distintos [35] [19], ou espelhos [32] [18] de maneira a que na mesma imagem se possa ter duas visões do aquário. É possível, ainda, extrair a posição 3D a partir de uma única imagem, utilizando fontes emissoras de luz para projetar a sombra do animal, e a partir delas inferir a posição do animal [38].

Técnicas que exploram o espaço 3D podem proporcionar uma reconstrução mais precisa da trajetória do animal, além de ser possível identificar situações de oclusão presentes em uma imagem pela análise de imagens provenientes de outros ângulos do cenário. Porém, o problema de utilizar várias câmeras ou espelhos está em como relacionar os objetos após identificá-los em cada câmera. Esse processo geralmente requer uma etapa de calibração trabalhosa [39]. A seguir, serão apresentadas algumas técnicas dessa categoria para o rastreamento de peixes e grupos de peixes.

Em 1996, Kato et al. [35] desenvolveu um sistema para extrair informações de peixes da espécie peixe-dourado utilizando duas câmeras, uma posicionada em frente e outra sobre o aquário, formando um ângulo de 90° entre elas. Dentre as informações extraídas, estão a velocidade, a direção e o deslocamento do animal no espaço 3D, utilizando apenas informações provenientes das imagens.

Basicamente os autores identificam o animal através de uma técnica de remoção de fundo aplicada em ambas as câmeras a partir de dois modelos de fundo, um para cada câmera. Uma vez identificado o peixe, sua posição no espaço tridimensional, no instante de tempo atual, é aproximada obtendo-se as coordenadas (x, z) da câmera de cima, e (x, y) da câmera frontal. É possível aproximar a posição dessa forma uma vez que a posição das câmeras formem um ângulo de exatamente 90°, caso contrário, se faz necessário um processo de triangulação, baseadas em Geometria Epipolar [40].

Os autores apresentam ainda um algoritmo para definir a direção de movimento do peixe a partir de três instantes de tempo consecutivos, denominados respectivamente de inicial(I), médio(M) e terminal(T). Inicialmente, calcula-se um círculo de raio r que inscreva esses três pontos e define-se a direção de movimento com base no diâmetro do círculo formado(Figura 4). Se o diâmetro do círculo formado for maior que a medida da diagonal máxima do aquário, considera-se que o animal mantém a mesma direção, caso contrário, a direção foi alterada para direita ou esquerda, de acordo com o ângulo formado pelos vetores \overline{IM} e \overline{MT} .

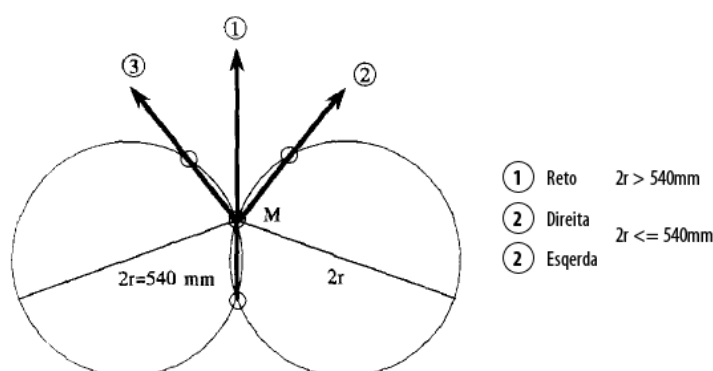


Figura 4 - Algoritmo para determinar direção do movimento de peixes.
 Fonte: Adaptado de Kato et al. [35]

Também focado na utilização de múltiplas câmeras, Wu et al. [19] propuseram um algoritmo para rastrear trajetórias de cada indivíduo em um grupo, como roedores, insetos e peixes através de duas câmeras. Assim como na abordagem de Kato et al. [35], os autores utilizam uma técnica de remoção de fundo para separar os animais do restante do cenário. A partir de imagens já segmentadas, o algoritmo proposto pode ser dividido em três etapas distintas. A primeira etapa consiste em utilizar o Filtro de Kalman para identificar cada indivíduo no cenário e associar as estimativas com a posição real do objeto dentre o conjunto de possíveis candidatos.

A segunda etapa consiste em estabelecer a correspondência entre os objetos detectados na imagem proveniente da primeira câmera com os objetos detectados na segunda. Para tanto, foi proposto um método de associação, baseado em geometria epipolar. A última etapa consiste em reconstruir trajetórias incompletas provenientes de erros de rastreamento. O algoritmo avalia os segmentos incompletos e procura interligá-los utilizando informações como tempo e continuidade do deslocamento.

Utilizando uma abordagem diferente das apresentadas até aqui [35] [19], Zhu et al. [18] propuseram um sistema que utiliza duas imagens, mas apenas uma câmera para o monitoramento de roedores e pequenos animais aquáticos. Na proposta, as imagens são obtidas a partir de espelhos colocados em posições específicas levando em consideração as características da câmera e posição do aquário, ou gaiola (Figura 5). Com esta configuração, a imagem capturada pela câmera possui a representação de três visões independentes do ambiente (Figura 6). Duas provenientes da reflexão da imagem do ambiente nos espelhos (Figura 6b e Figura 6c) e uma proveniente da visão direta da câmera. A abordagem também utiliza a geometria epipolar para calibrar as câmeras.

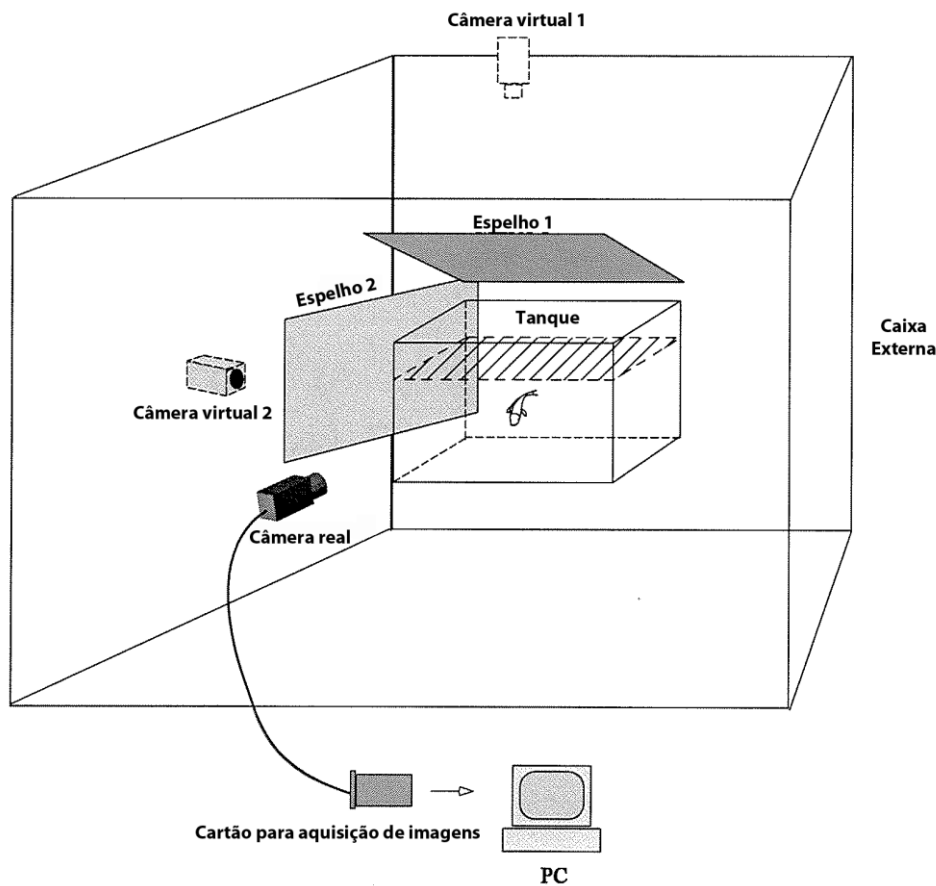


Figura 5 - Ilustração do aparato desenvolvido para o monitoramento dos animais com espelhos

Fonte: Zhu et al. [18]

No processo de obtenção da posição dos indivíduos, inicialmente é realizada uma segmentação, aplicando métodos de remoção de fundo e clusterização. Primeiramente, um algoritmo de remoção de fundo é utilizado para identificar regiões onde ocorre uma transição brusca de intensidades entre a imagem atual e o modelo de fundo. A seguir, um algoritmo de clusterização é empregado para agrupar pixels com intensidades próximas.

No contexto do trabalho de [18], cada cluster pode representar o corpo de peixes ou regiões demarcadas no corpo de pequenos roedores, por exemplo.

O rastreamento dos indivíduos é realizado através da análise de dois aspectos. O aspecto espacial em que se garante a correspondência entre os clusters nas três imagens e o aspecto temporal, em que se garante a correspondência destes ao longo dos *frames* da imagem, através de uma versão 3D do algoritmo de rastreamento de Kalman [41].

O sistema é capaz de extrair vários parâmetros relacionados aos animais identificados, como por exemplo, posição, deslocamento, velocidade, aceleração de cada animal, distância entre os animais, tempo gasto em uma região e tempo gasto em congelamento.

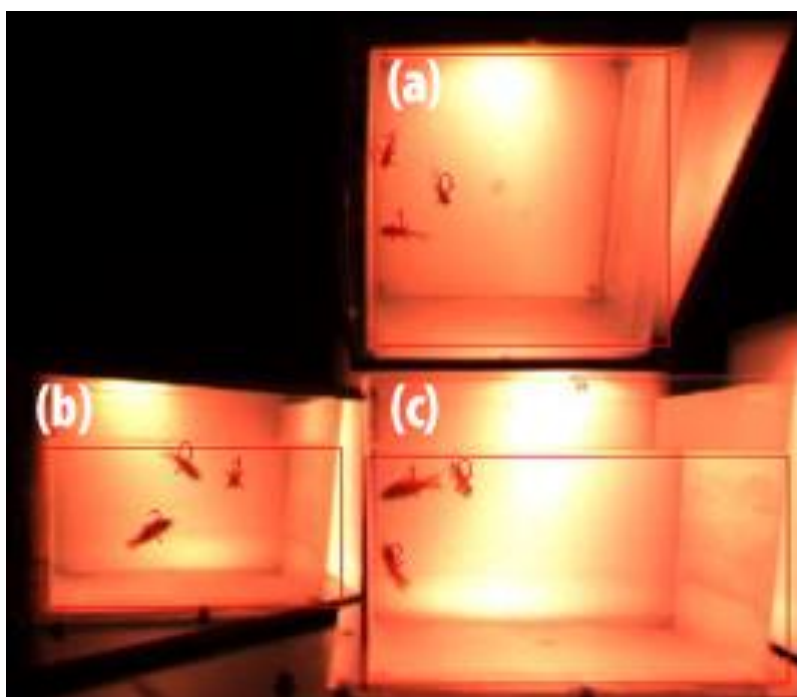


Figura 6 - Exemplo de Imagens provenientes das câmeras virtuais e da câmera real. (a) câmera virtual superior, (b) câmera virtual lateral, (c) câmera real.

Fonte: Adaptado de Zhu et al. [18]

3 RASTREAMENTO DE OBJETOS EM VÍDEO

Segundo Maggio [42], o processo de estimar a localização de um ou mais objetos de interesse, ao longo de uma sequência de imagens, é chamado de rastreamento em vídeo. O objetivo por trás do rastreamento é reconstruir a trajetória de alguns objetos através da obtenção de sua localização no espaço, a cada instante de tempo. Um bom método de rastreamento deve ser capaz de reconstruir esta trajetória o mais próxima da real possível.

O objeto de interesse pode variar de acordo com a aplicação. Por exemplo, em uma aplicação para o controle de fluxo em estradas, o objeto de interesse pode ser um carro, enquanto em uma aplicação de segurança, o objeto de interesse pode ser um indivíduo. Seja qual for a aplicação, é possível que o cenário sofra alterações ao longo do tempo devido ao deslocamento de outros objetos, à luminosidade e ao ruído proveniente do processo de digitalização de imagens, bem como a movimentações na câmera que capta os movimentos.

De acordo com Jalal [43], a arquitetura mais comum do processo de rastreamento em vídeo é formada por três módulos que interagem uns com os outros: modelagem, detecção e rotulagem do objeto. A *modelagem do objeto* é responsável por representar e caracterizar o objeto de interesse, baseando-se em características já conhecidas a cerca deste objeto. Cor, textura e geometria são algumas das diversas informações que podem ser utilizadas para definir um modelo. A *detecção de objetos* tem como objetivo identificar regiões candidatas onde o objeto de interesse pode ser encontrado. Algumas abordagens utilizam um modelo do objeto para varrer a imagem em busca dessas regiões em cada imagem de uma sequência, por exemplo, através da técnica de *template matching* [44] [45]. Já outras procuram explorar diferenças espaço-temporais entre imagens de uma sequência de forma a destacar regiões onde haja objetos em movimento [34].

Dessa forma, não é necessário um modelo de objeto, porém é necessário que o objeto esteja constantemente em movimento. Por fim, a *rotulagem do objeto* é responsável por identificar um objeto dentre um conjunto de candidatos, mantendo sua referência ao longo de uma sequência de imagens. A Figura 7 ilustra esta arquitetura bem como a interação entre seus módulos. As seções a seguir detalham as etapas de Modelagem, Detecção e Rotulagem do objeto.

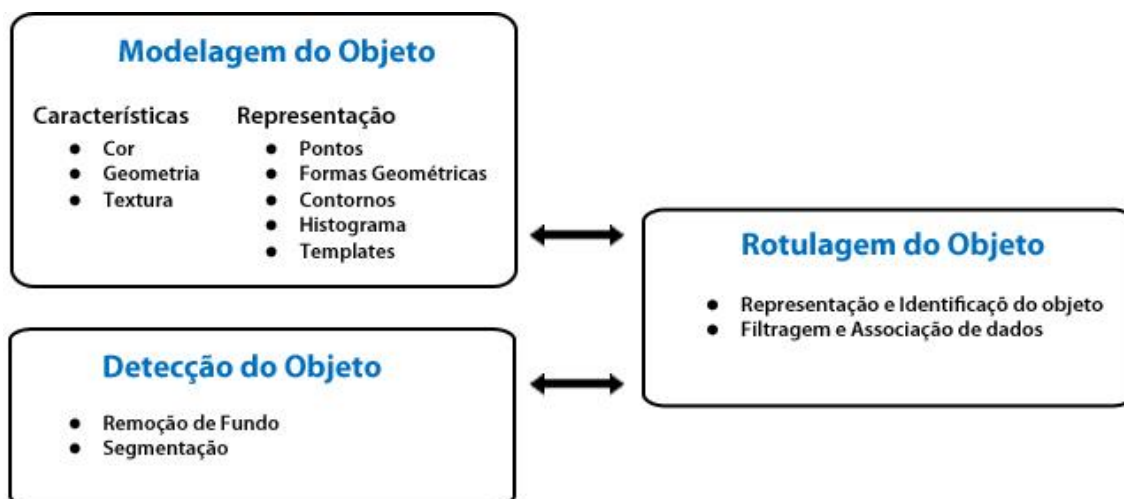


Figura 7 - Arquitetura para o rastreamento de objetos
Adaptado de: Jalal [43]

3.1 Modelagem do Objeto

A Modelagem do Objeto é uma etapa bastante importante no processo de rastreamento visual pois é nela que as informações responsáveis por caracterizar o objeto de interesse são definidas. Basicamente, um modelo de objetos define como um objeto é **representado** ao longo de uma sequência de imagens e quais **propriedades** o caracterizam.

3.1.1 Representação

Um objeto pode ser representado, dentre outras formas, por uma ou mais das seguintes informações: pontos, formas geométricas primitivas, contornos, blobs, bordas, templates, histogramas, dentre outros. Um *ponto* é a forma mais primitiva existente em imagens digitais. Um objeto é representado por um pixel da imagem, que simboliza, por exemplo, o centro geométrico desse objeto [46] ou por um conjunto de pixels de interesse no corpo do objeto [47].

Devido à simplicidade do ponto, muitas vezes se faz necessária uma forma mais avançada de representação, capaz de fornecer maiores informações sobre o objeto. Para essa finalidade, *formas geométricas primitivas* como retângulos e elipses são utilizadas. O retângulo, Figura 8a, muitas vezes é utilizado para representar carros [48] e pessoas [49]. A elipse (Figura 8b) também pode ser usada para representar esses mesmos objetos [50] [51].



Figura 8 - Exemplos de representação de objetos através de formas geométricas primitivas (a) e (b) e contornos (c).

Adaptado de: Osuna [49], Xia [50] e Song [52].

Um *contorno*, Figura 8c, nada mais é do que a representação dos limites de um objeto. Esta é uma forma bastante útil para representar objetos não rígidos, pois permite uma descrição ainda mais detalhada que formas geométricas primitivas, como por exemplo o perímetro desse objeto [53] [52]. Um *blob*, ou *silhueta*, Figura 9, corresponde a região interna ao contorno e também é uma forma muito comum de representação. Através de um blob, a área do objeto pode ser facilmente determinada.

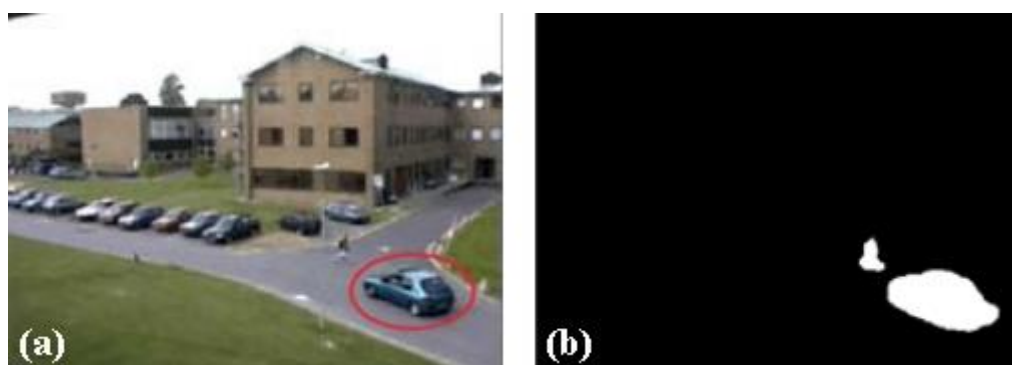


Figura 9 - Representação de objetos através de blobs. (a) imagem de um cenário. (b) objetos de interesse segmentados.

Adaptado de: [54].

O *template* é um modelo contendo um conjunto de características que representam o objeto de interesse. Um template pode ser uma imagem, um conjunto de formas geométricas ou os contornos do objeto em questão. A técnica mais conhecida baseada em templates é o *template matching* [55], que consiste em varrer uma imagem em busca de regiões que mais se assemelham ao template utilizado, através de medidas de similaridade como Soma das Diferenças Absolutas(SAD) [44] e Somados Quadrados das Diferenças(SSD) [45]. Um exemplo de template e da imagem a ser analisada podem ser vistos na Figura 10.



Figura 10 - Representação de objetos através de templates.
Adaptado de: Alsaade [44]

Um objeto pode, ainda, ser representado pelo seu *histograma* de cores, ou de níveis de cinza [55]. O histograma nada mais é que uma representação da distribuição dos pixels de uma imagem. Visualmente, um histograma pode ser ilustrado como um gráfico, em que a barra horizontal representa a variação de tonalidades, enquanto a barra vertical representa a quantidade de pixels para cada tonalidade. A maior vantagem de representar um objeto através de seu histograma, é que independente da orientação do objeto, o histograma permanece quase inalterado. Uma desvantagem desta abordagem é que objetos completamente diferentes podem possuir a mesma distribuição de intensidades, pois o histograma não contém informação referente à distribuição espacial destas intensidades, apenas a sua frequência. A Figura 11 ilustra uma imagem em tons de cinza acompanhada do seu histograma.

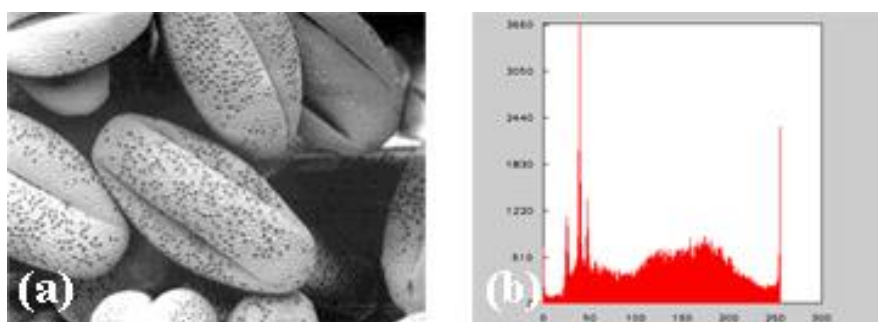


Figura 11 - Representação de objetos através de histogramas.
Adaptado de: Gonzalez [55].

3.1.2 Propriedades

Um passo importante na etapa de Modelagem do Objeto é escolher quais propriedades serão utilizadas para diferenciar um objeto do restante do cenário. Dentre as propriedades visuais mais comuns em imagens digitais, estão: bordas, cor e textura.

As *bordas* ou *limites* de um objeto geralmente são acompanhadas por mudanças bruscas de intensidade e mesmo em condições de luminosidade ruins, esta propriedade se

mantém. A representação baseada em contornos, por exemplo, geralmente utiliza bordas como propriedades. Existem diversos algoritmos capazes de extrair os limites de um objeto, os mais populares são o Operador de Sobel [56] e o detector de bordas Canny [57].

A *cor* é uma das características visuais mais intuitivas para se descrever um objeto, sendo capaz de discriminá-lo até mesmo em casos de oclusão parcial [58]. A representação baseada em histogramas, por exemplo, utiliza a intensidade de cor de cada pixel como propriedade. Entretanto, diferentemente das bordas, a cor é muito mais sensível a variações de luminosidade, ao tipo de ambiente e do objeto de interesse.

A *textura* é uma propriedade que descreve a variação de intensidades de cor de um conjunto de pixels vizinhos. A representação por templates, por exemplo, pode utilizar a textura do objeto de interesse ao invés do próprio objeto como modelo, possibilitando um certo grau de resistência à variações de postura. Para utilizar a propriedade de textura, é necessário definir um descritor que servirá para quantificar propriedades como suavidade, regularidade e rugosidade de uma superfície. De acordo com Gonzalez, os principais descritores de textura são baseados em abordagens estatísticas, estruturais e espectrais [55].

3.2 Detecção do Objeto

A detecção do objeto é a etapa responsável por identificar a região onde o objeto de interesse pode ser encontrado utilizando um conjunto de características conhecidas desse objeto. Segundo Yilmaz [59], é muito comum utilizar apenas informação do *frame* corrente para detecção, embora alguns métodos utilizem informações obtidas em uma sequência de imagens anteriores, de forma a reduzir falsos positivos. Os métodos mais comuns para realizar a detecção de objetos são baseados em remoção de fundo [60] e segmentação [61].

A remoção de fundo é uma técnica amplamente utilizada para detectar objetos móveis em vídeos [62]. O objetivo é aplicar a diferença entre a imagem corrente, I_c , e uma referência, I_r , e gerar uma imagem binária resultante, I_d , na qual um valor representa o fundo e outro representa o objeto de interesse, conforme a equação:

$$I_{d(x,y)} = \begin{cases} I_{c(x,y)} = I_{r(x,y)} & , 1 \\ I_{c(x,y)} \neq I_{r(x,y)} & , 0 \end{cases} \quad (1)$$

A referência, comumente chamada de imagem de fundo, deve conter apenas a representação do cenário em que não há objetos móveis. Logo, é necessário obter um modelo de fundo inicial. Este modelo pode ser capturado quando os objetos não estiverem na cena, ou pode ser gerado a partir da média de intensidades em um conjunto de imagens [60]. A principal vantagem do primeiro modelo é que, em muitos casos, ele é mais facilmente obtido, porém não é muito útil em aplicações sujeitas a variações de iluminação. Já o segundo se adapta muito bem a variações de iluminação e ruídos, embora seja mais computacionalmente custoso. Este último muitas vezes é referenciado como modelo de fundo adaptativo.

Assim como as técnicas baseadas em remoção de fundo, técnicas baseadas em segmentação também buscam particionar a imagem em regiões, porém, a diferença entre estas duas abordagens está no significado dessas regiões. Enquanto a remoção de fundo divide a imagem em objetos móveis e fundo da cena, a segmentação divide a imagem em regiões similares, ou regiões que compartilhem um dado conjunto de características. Dessa forma, se uma aplicação deseja detectar automóveis em um estacionamento, a remoção de fundo não faz distinção entre carros, pessoas, animais e aviões por exemplo. Por outro lado, técnicas baseadas em segmentação utilizam um conjunto de informações que descrevem o objeto de interesse para particionar apenas as regiões que mais se assemelham a este objeto. Arbeláez et al. [61], por exemplo, propôs um algoritmo para segmentação de imagens baseado na análise e hierarquização de contornos. Basicamente, o algoritmo é responsável por avaliar os contornos encontrados em uma imagem e agrupá-los em diferentes hierarquias, nas quais cada hierarquia representa um objeto diferente da cena.

3.3 Rotulagem do Objeto

Uma vez definido o conjunto de candidatos, a etapa de rotulagem é responsável por realizar a correspondência de um objeto identificado na imagem anterior com um candidato na imagem corrente. Algumas das abordagens mais populares para esta tarefa baseiam-se em Estimativas de Posição e em Busca por Características. As seções a seguir apresentam estas abordagens.

Abordagens baseadas em estimativas necessitam de informação prévia do cenário e do objeto, e lidam com a avaliação de diferentes hipóteses sobre as variáveis de interesse como posição, velocidade e aceleração, por exemplo. O principal algoritmo que utiliza esta

abordagem é o Filtro de Kalman [41] que utiliza um conjunto de medidas observadas ao longo do tempo para produzir estimativas a cerca de variáveis desconhecidas, levando em conta o possível ruído dessas observações. Essas variáveis podem ser compostas por um conjunto de informações que se deseja avaliar como, por exemplo, posição e velocidade de um determinado objeto. Para o Filtro de Kalman, esse conjunto de variáveis é denominado de **estado** e geralmente é referenciado como X_t .

De acordo com Welch [63], o processo de utilização do Filtro de Kalman pode ser dividido em duas etapas: predição e correção. O primeiro passo é a geração de um estado inicial para as variáveis de interesse, denominado X_t^- . Com base em uma medida observada, z_t , é gerada a estimativa corrigida, denominada X_t^+ . No tempo seguinte ($t + 1$), esta estimativa é usada como estimativa inicial e o processo recomeça. A estimativa inicial é comumente chamada de estado *a priori*, enquanto que a estimativa corrigida é chamado de estado *a posteriori*. No contexto de processamento de imagens, a medida observada (z_t) pode ser gerada através do centro do blob, obtido por um processo de segmentação de imagens, por exemplo.

Uma propriedade importante do Filtro de Kalman é o Ganho de Kalman. O Ganho de Kalman é uma função que procura minimizar a matriz de covariância do erro estimado. Quando o ganho é alto, o filtro confia mais nas medidas observadas que em sua estimativa inicial, e por consequência, corrige essa estimativa aproximando-a da observação. Por outro lado, quando o ganho é baixo, a correção não diverge muito da estimativa inicial.

É necessário ainda, definir um conjunto de informações básicas adicionais para que o método seja capaz de estimar as variáveis de interesse. Este conjunto de informações pode ser visualizado na Tabela 2.

Ao contrário de abordagens baseadas em estimativas, abordagens baseadas na busca por características necessitam avaliar cada imagem para identificar o objeto, muitas vezes em cenários complexos. Para tanto, é definido um conjunto de características que é utilizado para diferenciar o objeto de interesse do restante do cenário. Os principais algoritmos nessa categoria, descritos na literatura, são CAMShift [64] e SURF [65].

Tabela 2 - Conjunto de informações que compõem o filtro de Kalman para Rastreamento

Simbolo	Descrição	Exemplo
A_t	Modelo que descreve a transição de estado no instante de tempo t	Equação do movimento retilíneo uniformemente variado(MRUV)
H_t	Modelo que descreve as observações no instante de tempo t	Algoritmo que obtém a posição de um objeto em movimento na cena, o qual é um candidato ao objeto de interesse
Q_t	Covariância que descreve o ruído no processo de estimativa no instante de tempo t	Percentual de confiança na estimativa obtida pelo modelo
R_t	Covariância que descreve o ruído na medida observada no instante de tempo t	Percentual de confiança na estimativa obtida pela observação

Proposto por Gary Bradski em 1998, o CAMShift é uma expansão do algoritmo *mean-shift* para tratar distribuições de probabilidade capazes de mudar ao longo do tempo [64]. No contexto de processamento de imagens, a distribuição de probabilidades é representada através do histograma do objeto de interesse, que pode mudar de acordo com o deslocamento do objeto na cena ou, também, devido a características extrínsecas ao objeto como, por exemplo a iluminação da cena, o ruído presente na sequência de imagens e o processo de digitalização.

Inicialmente, o método foi proposto para a detecção de faces em imagens utilizando apenas a informação referente à matiz, ignorando os outros canais do espaço de cores HSV(matiz, brilho e valor). O método, entretanto, pode ser utilizado com os outros canais [66].

Em linhas gerais, o algoritmo começa obtendo o histograma de uma janela retangular centrada na posição inicial do objeto. Na imagem subsequente, o algoritmo expande a região de busca e procura pela janela de maior semelhança local com o histograma e move a região de busca para o centro desta janela. Este processo é repetido até que o pico de semelhança seja obtido ou um limite pré definido de iterações seja atingido. O algoritmo foi elaborado de forma que a janela de busca seja ajustada a cada iteração para suportar mudanças no tamanho e orientação do objeto de interesse.

O *Speed Up Robust Feature* (SURF) é um método desenvolvido pelo pesquisador Herbert Bay para a detecção de pontos de interesse em uma imagem e extração de um vetor de características que descreve estes pontos [65]. Dessa forma, extraindo-se um vetor

de características do objeto e outro vetor do cenário, o objetivo é encontrar o objeto na cena através da identificação, no vetor do cenário, das características do vetor do objeto [67]. Uma das grandes vantagens desse método é sua capacidade de encontrar o objeto de interesse mesmo que esse se encontre em orientação e escala distintas das extraídas inicialmente.

Para a definição dos pontos de interesse, o detector busca na imagem por informações como blobs e fronteiras, através de operadores baseados na divergência do gradiente de uma imagem, como a aproximação do Laplaciano da Gaussiana (LoG) pela Diferença de Gaussianas (DoG) [68] por exemplo, com a vantagem de encontrar o espaço-escala [69] da imagem, propriedade necessária para tratar variações no tamanho de objetos. Os pontos que se destacam nos blobs e fronteiras detectados, são escolhidos como pontos de interesse. Para cada ponto de interesse detectado, o SURF aplica a transformada de Haar D [70] nos eixos horizontal (D_x), vertical (D_y) e constrói um vetor que os descreve. Ao final desse processo, o conjunto de vetores obtido é agrupado em um vetor final e, a partir deste ponto, este último estará pronto para ser comparado com um outro vetor com o objetivo de relacionar pontos de interesse correspondentes.

4 AVALIAÇÃO DE MÉTODOS DE RASTREAMENTO APLICADOS A PEIXES

A fim de avaliar a acurácia dos métodos apresentados na seção 3.3, no contexto de rastreamento de cardumes, foram conduzidos testes em dois cenários distintos. Estes cenários foram obtidos no Instituto de Toxicologia e Farmacologia da Pontifícia Universidade Católica do Rio Grande do Sul.

O primeiro cenário utilizado contém apenas um peixe em um aquário e um segundo, dez peixes. Em ambos os experimentos, foi utilizada uma câmera capturando imagens com resolução de 720x640, a uma taxa média de 30 quadros por segundo, por um período de aproximadamente 5 minutos.

As imagens das Figura 12 ilustram alguns quadros retirados do vídeo com um peixe e com múltiplos peixes. No primeiro cenário, uma câmera foi posicionada aproximadamente 50cm a frente de um aquário. Este aquário possui um peixe adulto da espécie peixe-zebra que mede cerca de 3 a 4 centímetros de comprimento e possui coloração escura. Devido ao posicionamento da câmera e do vidro na parte frontal no aquário, é possível ver o reflexo de um pesquisador em algumas imagens.



Figura 12 - Imagens retiradas do primeiro cenário

Já no segundo cenário, a câmera está posicionada cerca de 50cm acima de um aquário com 10 peixes adultos da espécie peixe-zebra e de coloração ainda mais escura que os anteriores. Neste cenário, não há reflexos na região frontal da imagem pois o aquário é aberto em sua parte superior.



Figura 13 - Imagens retiradas do segundo cenário

Para realizar os testes, foram utilizadas as implementações existentes na biblioteca OpenCV versão 2.4.6 para cada um dos métodos apresentados no capítulo 3.3. Nas seções 4.1.1 , 4.1.2 e 4.1.3 são apresentados, respectivamente, testes utilizando o método CAMShift, Filtro de Kalman e SURF.

4.1.1 CAMShift

De acordo com a implementação do método CAMShift na biblioteca OpenCV, o histograma do objeto é obtido na primeira iteração do algoritmo. Logo, a janela fornecida inicialmente deve estar centrada no objeto de interesse. Buscando reduzir problemas de variação de luminosidade e ruído, a implementação usada como referência agrupa as intensidades dos histogramas em 16 níveis distintos. Para comparar a similaridade dos histogramas, foi utilizado o operador de correlação linear,

$$d(H_1, H_2) = \frac{\sum_i (H_1(i) - H_1^-)(H_2(i) - H_2^-)}{\sqrt{\sum_i (H_1(i) - H_1^-)^2 \sum_i (H_2(i) - H_2^-)^2}} \quad (2)$$

em que, H_1 e H_2 representam os histogramas do peixe e do cenário respectivamente e i representa uma intensidade do histograma. O operador de correlação utilizado não é necessariamente o mesmo utilizado na implementação do CAMShift, este é utilizado apenas para fins ilustrativos.

Inicialmente, os histogramas foram construídos utilizando a Matiz do espaço de cores HSV, conforme a proposta original do método, o que não apresentou resultados satisfatórios. Acredita-se que estes resultados se devem, principalmente, à baixa qualidade das imagens utilizadas e a coloração escura do animal. Dessa forma, optou-se por utilizar o canal Valor ao invés da Matiz para produzir os histogramas, o que produziu melhores resultados, conforme descrito a seguir.

No teste realizado com um único peixe, o CAMShift apresentou resultados positivos para identificar o animal e manter a correspondência ao longo do experimento. Conforme se pode observar na Figura 14, ocorre uma predominância de índices de correlação acima de 80% e, mesmo quando o índice é baixo, por exemplo em T90, as iterações seguintes tendem a convergir sempre ao pico de similaridade.

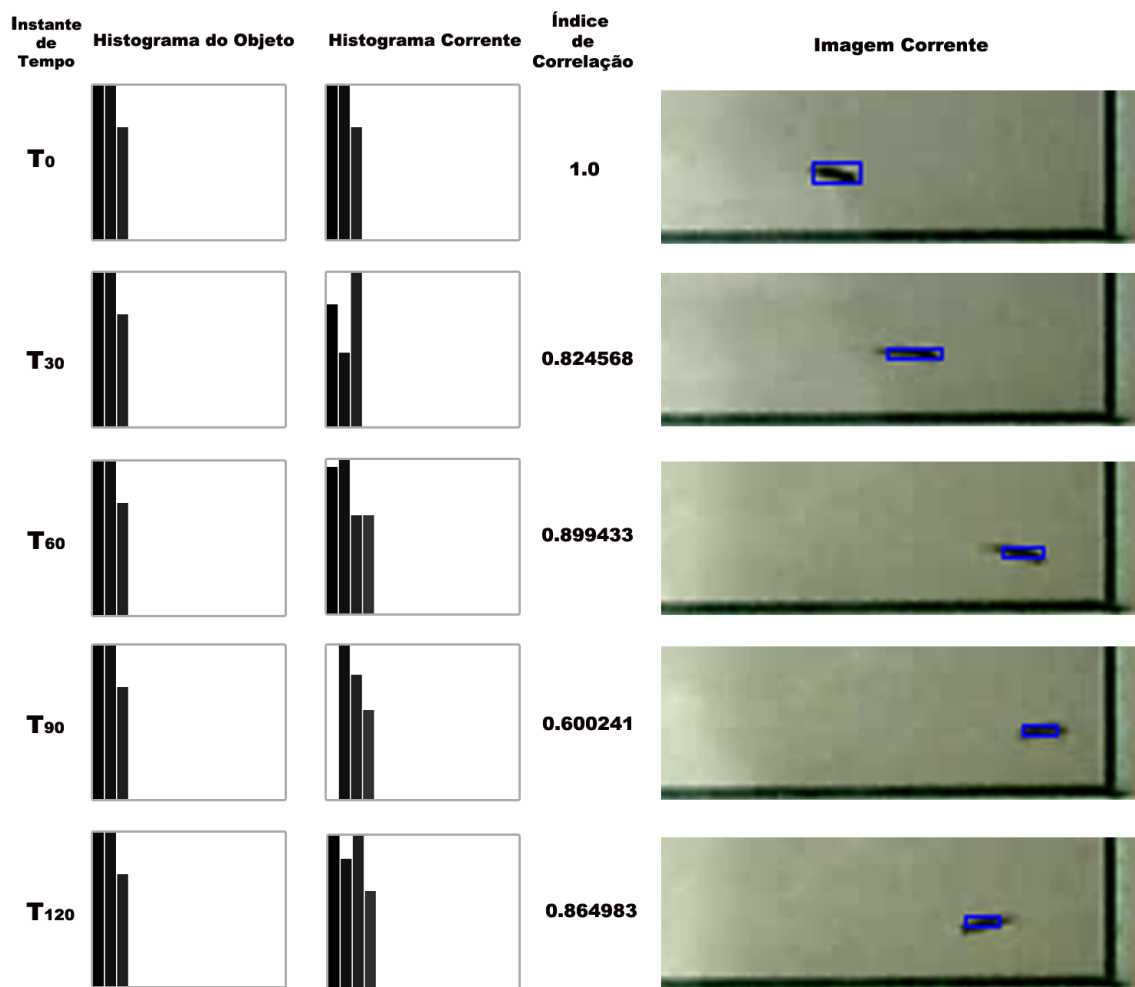


Figura 14 - Conjunto de dados obtidos através do método CAMShift para rastrear um único peixe.

Após a execução do teste com um peixe, foram realizados testes com o vídeo de múltiplos peixes. Estes testes não apresentaram resultados bons, pois a janela de busca tende a divergir do animal quando um segundo animal se aproxima, como pode ser visto na Figura 15.

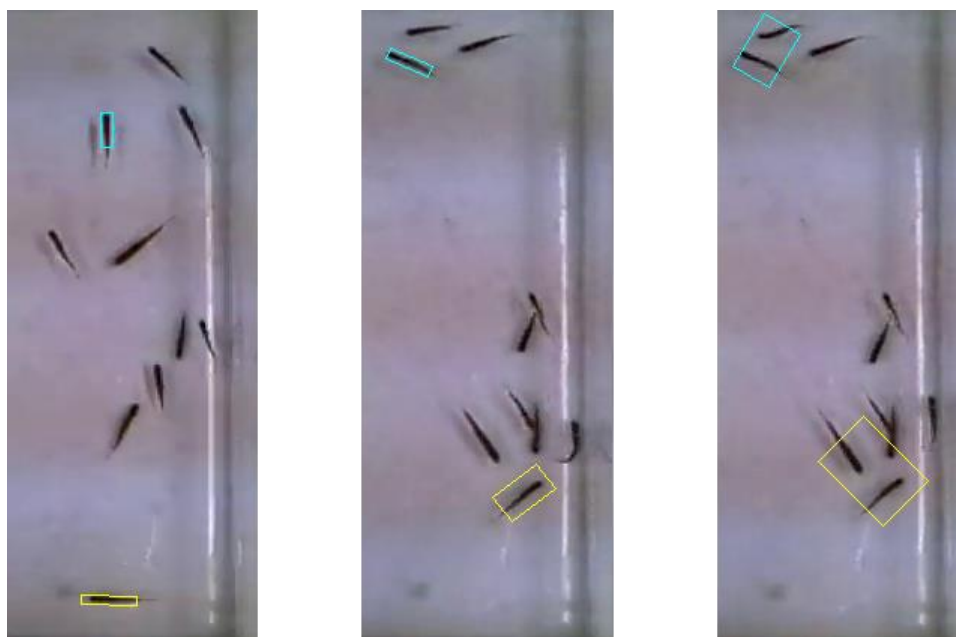


Figura 15 - Ilustração da janela de busca divergindo quando outros animais se aproximam.

4.1.2 Speed-Up Robust Features

Como mencionado na seção 3.3 o método SURF consiste em uma etapa de detecção de regiões de interesse e uma etapa de extração de um vetor de características do objeto de interesse e um do cenário. Para obter o conjunto de regiões de interesse e o vetor de características do animal em cada cenário, foram obtidas duas imagens, uma para cada cenário, nas quais aparece somente um peixe, como pode ser visto na Figura 16. Já as informações referentes ao cenário são obtidas em cada imagem da sequência.

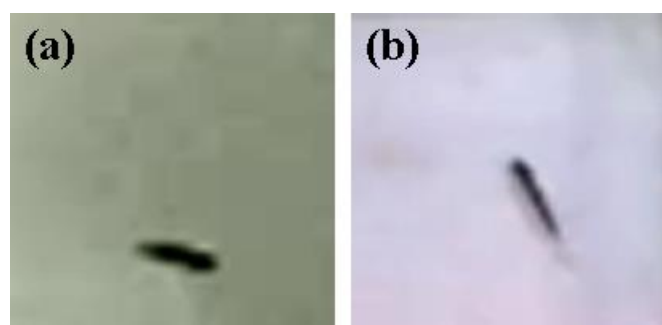


Figura 16 - Imagens capturadas de peixes em ambos os cenários. (a) cenário com um peixe, (b) cenário com múltiplos peixes

Os testes iniciais conduzidos com o método SURF apresentaram resultados ruins para o cenário com um único peixe. Observou-se que estes resultados se dão, em grande parte, devido às características naturais do animal em questão. Como o peixe-zebra é um animal pequeno e de coloração bastante escura, o detector não é capaz de extrair uma

quantidade suficiente de regiões de interesse para identificar o animal na primeira etapa. Nas imagens da Figura 17, por exemplo, é possível observar que nenhuma das regiões detectadas são referentes ao animal e, conseqüentemente, o vetor de características extraído não é capaz de detectá-lo. Visto que o teste com um peixe não apresentou bons resultados, não foram realizados testes no cenário com múltiplos peixes.

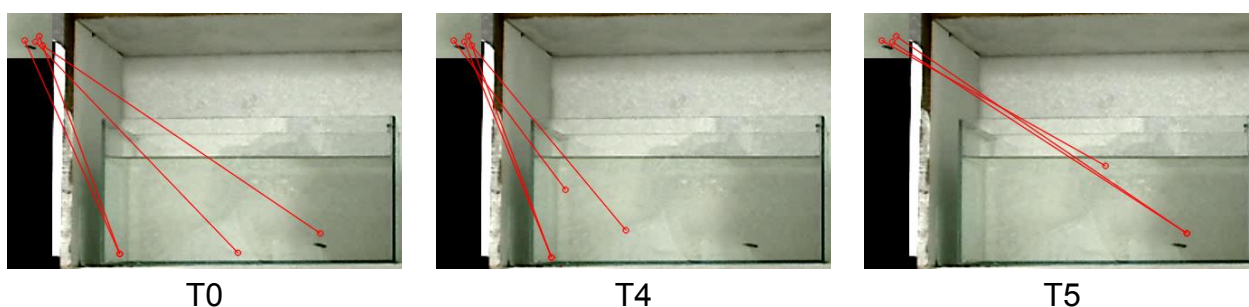


Figura 17 - Conjunto de imagens obtidas através do método SURF para o rastreamento de um peixe.

4.1.3 Filtro de Kalman

Como mencionado na seção 3.3, o filtro de Kalman é composto por um conjunto de equações capazes de inferir o estado de variáveis desconhecidas com base em observações realizadas ao longo do tempo. Para tanto, deve-se definir o conjunto de variáveis que se deseja avaliar, o método para obtenção das medidas observadas e as informações apresentadas na Tabela 2.

Por se tratar de rastreamento, foram escolhidas como variáveis para compor o modelo, as seguintes grandezas: a posição nos eixos x, y do plano cartesiano e a velocidade linear em ambos os eixos,

$$X_t = \{p_{xt}, p_{yt}, v_{xt}, v_{yt}\}. \quad (3)$$

na qual p_{xt} , p_{yt} representam a posição do animal no instante de tempo t , p_{xt} e p_{yt} representam sua velocidade e X_t representa o estado do peixe.

Para a obtenção das medidas observadas foram avaliadas duas abordagens baseadas em remoção de fundo: a primeira consiste em uma técnica de remoção de fundo por diferença de *frames* e, a segunda, a mesma técnica porém, realçada através de um operador laplaciano. Na Figura 18, são ilustrados blobs capturados em imagens contendo diversos peixes em um aquário, após a aplicação das duas técnicas de segmentação

mencionadas. É possível observar que a segunda técnica produz blobs mais precisos, sendo por isto, a técnica escolhida para a obtenção das medidas observadas.

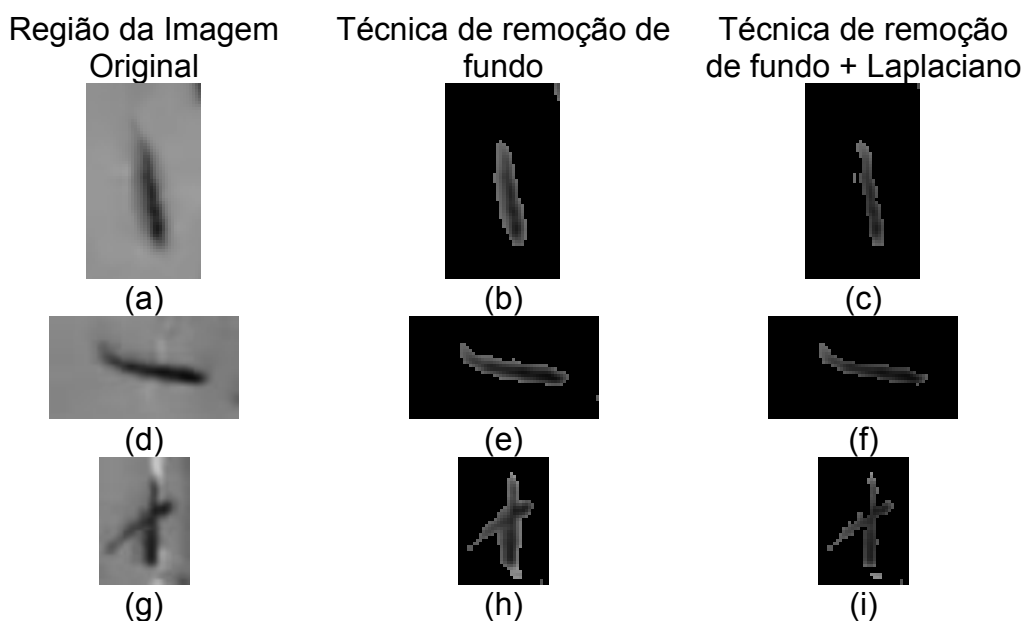


Figura 18 - Comparação entre blobs obtidos por duas técnicas distintas.

Após a definição do método para obtenção das medições, deve-se ainda definir as informações apresentadas na Tabela 2. Dadas as variáveis de interesse, buscou-se um modelo de transição que seja capaz de inferir a posição futura do animal com base em sua posição e velocidade no passado, além de manter sua velocidade constante. O seguinte modelo de transição foi utilizado:

$$A_t = \begin{cases} p_{xt} & = p_{x,t-1} + v_{x,t-1} \\ p_{yt} & = p_{y,t-1} + v_{y,t-1} \\ v_{xt} & = v_{x,t-1} \\ v_{yt} & = v_{y,t-1} \end{cases} \quad (4)$$

Já o modelo de observação, consiste nas medidas observadas obtidas através do método de segmentação,

$$H_t = \begin{cases} p_{xt} & = Zp_{xt} \\ p_{yt} & = Zp_{yt} \\ v_{xt} & = 0 \\ v_{yt} & = 0 \end{cases}, \quad (5)$$

em que Zp é um ponto que representa um determinado blob.

Por fim, deve-se definir o possível ruído que as estimativas de ambas as etapas de predição e correção podem produzir. Nos testes realizados, adotou-se um percentual de confiança de 95% tanto na estimativa inicial quanto na medida observada.

Após a definição das informações mencionadas, o filtro de Kalman foi aplicado ao cenário com um peixe, apresentando estimativas precisas da posição do animal. Isto se deve, em grande parte, aos dados da observação, provenientes do método de detecção de blobs, que é capaz de separar o peixe do restante do cenário muito claramente e consequentemente fornecer observações precisas da posição do animal. Na Figura 19, são apresentadas imagens obtida após executar o filtro de Kalman em diferentes instantes de tempo.

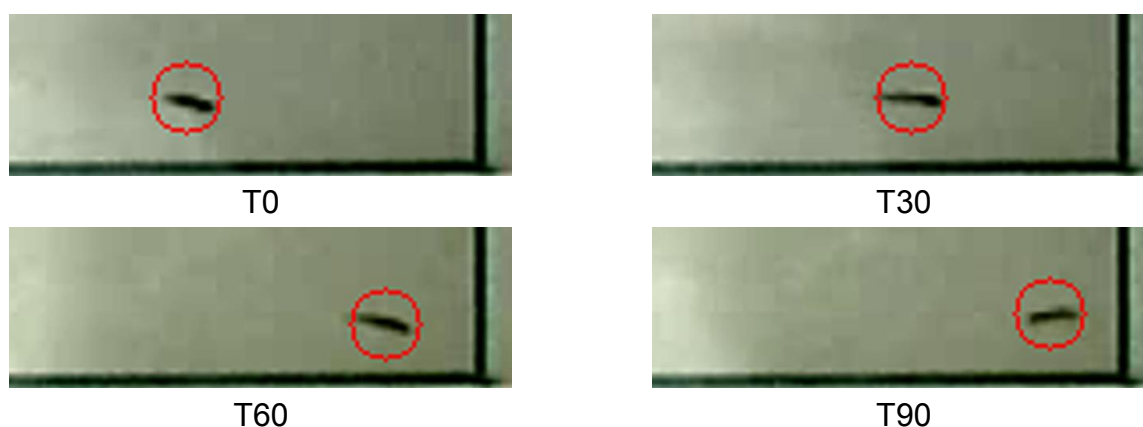


Figura 19 - Conjunto de dados obtidos através do filtro de Kalman para rastrear um único peixe.

A seguir foi realizado o teste do filtro de Kalman com o vídeo do aquário com 10 peixes. Para tanto, é necessário modificar o algoritmo que fornece a observação da próxima posição do peixe que está sendo rastreado. No cenário com um único peixe o processo de segmentação gera apenas 1 blob, cuja posição central é usada como a observação. No caso de um aquário com 10 peixes, esta segmentação poderá gerar até 10 blobs distintos e um deles deverá ser escolhido como sendo a nova observação. Para esta escolha utilizou-se como métrica o blob mais próximo da estimativa de posição para o quadro atual.

Após o ajuste no algoritmo, foram realizados testes do filtro de Kalman com múltiplos peixes. Os resultados obtidos neste segundo cenário não foram satisfatórios, pois a referência de um peixe é ser perdida quando um ou mais peixes se cruzam. Na Figura 20, este problema pode ser visualizado com o peixe de marcador circular nos instantes de tempo T5 e T13 e com o peixe de marcador retangular nos instantes de tempo T13 e T17.

Apesar deste problema, é possível notar um aspecto interessante no uso do filtro de Kalman, nos instantes de tempo T0 e T8. O peixe representado pelo marcador retangular mantém a correta identificação do animal mesmo com peixes muito próximos.

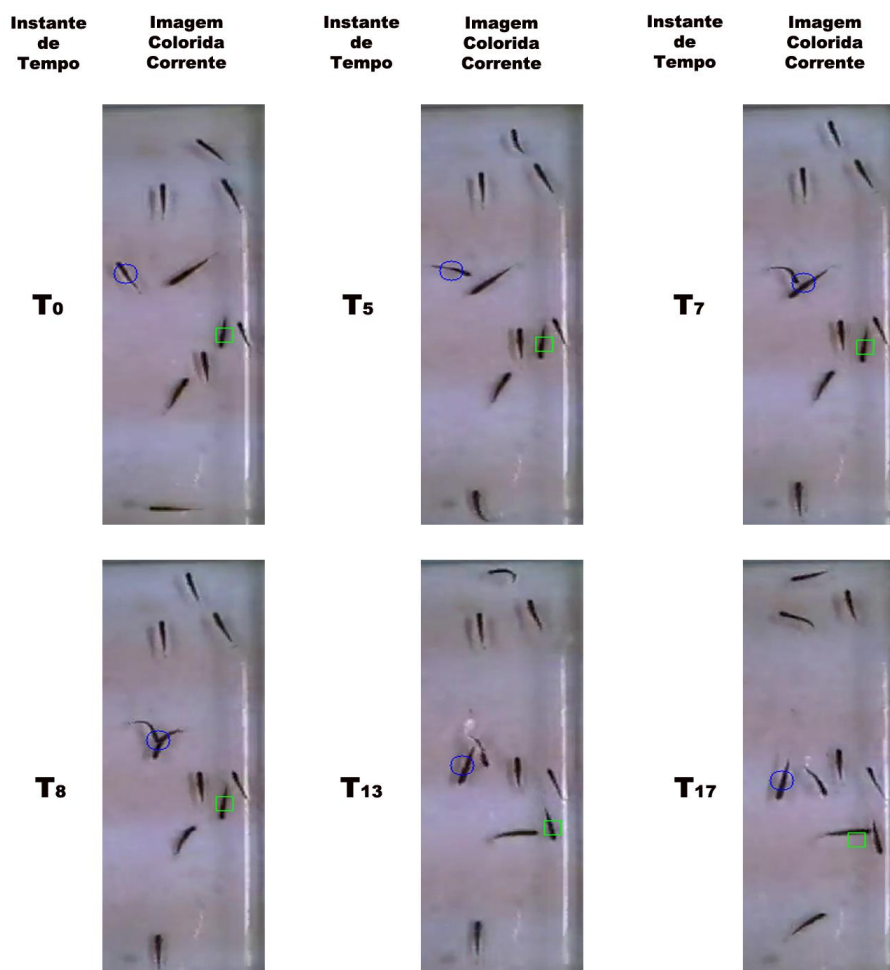


Figura 20 - Conjunto de imagens obtidas através do filtro de Kalman para rastrear múltiplos peixes

Considerando que o problema relacionado a perda de referência pode estar associado a métrica que escolhe a observação correta para cada instância do filtro, é possível adaptar esta métrica para utilizar a estimativa do filtro em questão ao invés de corrigi-la com uma observação imprecisa. Na Figura 21, é possível observar que após esta adaptação, o peixe de marcador circular, ainda assim, perdeu sua referência a partir de T7. Em contrapartida, o peixe de marcador retangular mantém sua referência desde o início, diferentemente do que ocorria no instante de tempo T17 da Figura 20.

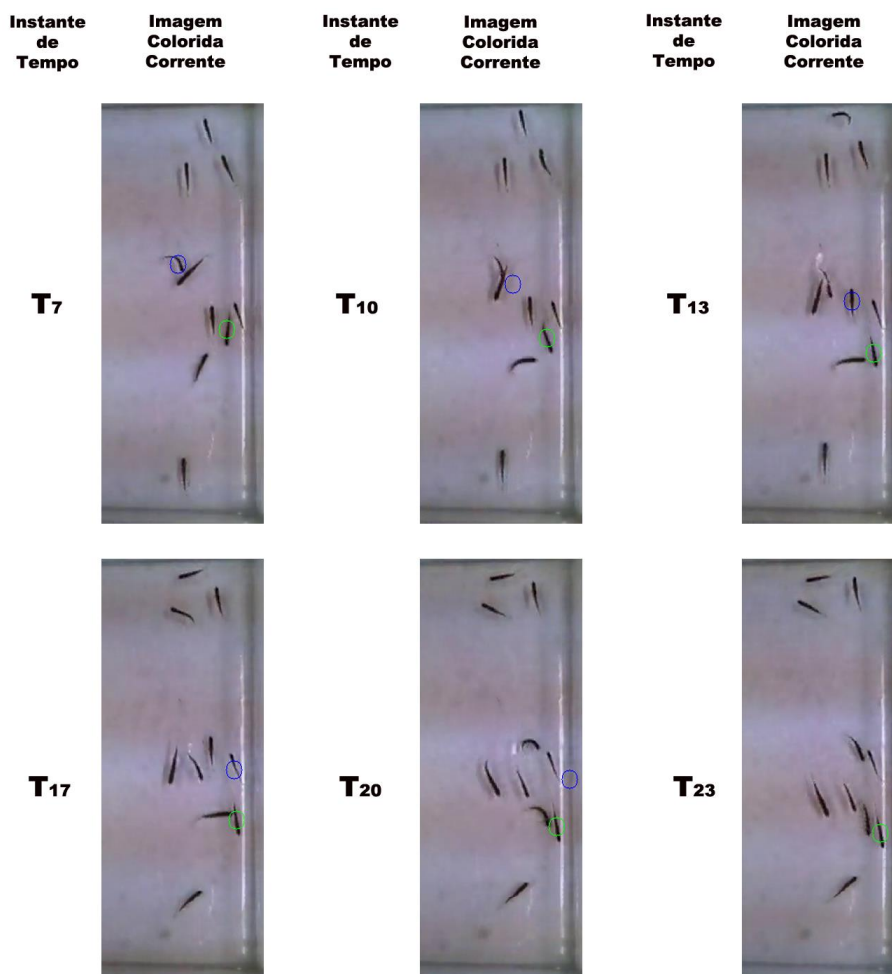


Figura 21 - Conjunto de imagens obtidas após adaptação do Filtro de Kalman para o rastreamento de múltiplos animais.

4.1.4 Análise dos Resultados

Nos testes apresentados, foi avaliada a acurácia dos métodos de rastreamento e observou-se que, todos os três métodos sofreram com o problema de oclusão entre peixes. Os métodos baseados em intensidades de cor (CAMShift e SURF), por exemplo, não apresentaram resultados satisfatórios, sendo que o SURF apresentou resultados ruins mesmo com o aquário de um único peixe. O Filtro de Kalman, por outro lado, apresentou melhores resultados, embora não tenha sido capaz de manter a identidade dos peixes por muitos *frames*.

Os resultados ruins do método SURF se devem, em grande parte, à baixa quantidade de detalhes na coloração da espécie de peixes utilizada nos testes. Já o método CAMShift, obteve resultados bons no primeiro cenário, porém, na presença de mais animais no aquário, o método não foi capaz de manter a identidade do peixe por muitos *frames*. Neste cenário, observou-se que a janela de busca do algoritmo não era corretamente

redimensionada quando outros animais se aproximavam. Além disso, observou-se que a janela de busca deveria ser mais compacta de forma a minimizar o risco de falsos positivos.

Por estas razões, não foi possível observar o potencial das técnicas CAMShift e SURF nos testes realizados. Como estas técnicas se baseiam nas intensidades das imagens, o ruído presente nas imagens utilizadas e a possibilidade dos animais ficarem próximos uns dos outros tiveram um impacto muito significativo na precisão dos métodos.

O Filtro de Kalman apresentou resultados satisfatórios no primeiro cenário. Já no cenário de múltiplos peixes, os resultados obtidos não foram bons, pois as observações obtidas nem sempre apontavam para o indivíduo correto, o que degrada o desempenho do método. Entretanto, o Filtro de Kalman mostrou-se uma abordagem bastante interessante, pois a precisão da técnica independe da qualidade das imagens. Logo, ao aumentar a confiabilidade destas observações, a técnica tende a apresentar resultados mais precisos.

No que diz respeito à geração das observações, o maior desafio encontrado nos testes iniciais foi o tratamento de oclusão no cenário com vários peixes pois, como o peixe-zebra é uma espécie naturalmente ativa em cardumes, os peixes tendem a se cruzar com bastante frequência ao longo do experimento. Nas imagens da Figura 22, por exemplo, é possível observar algumas ocorrências de oclusão retiradas do vídeo com múltiplos peixes. Algumas vezes, como na Figura 22c, o cruzamento se dá por quatro peixes, produzindo uma diferença entre a quantidade de estimativas e a quantidade de observações. Essa diferença faz com que haja mais de uma estimativa apontando para um mesmo blob na imagem.

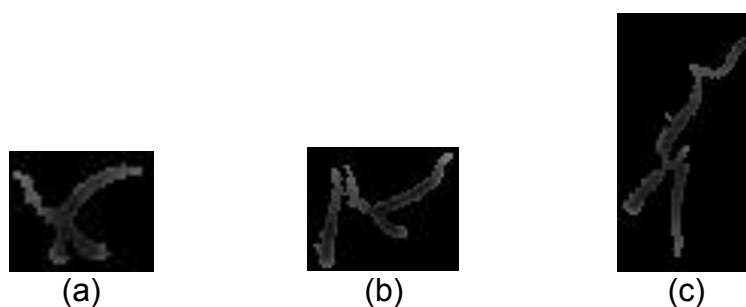


Figura 22 - Exemplo de blobs em que há oclusão ou sobreposição entre peixes no aquário.

5 MÉTODO DESENVOLVIDO

Os métodos de rastreamento que utilizam abordagens 2D, em trabalhos anteriores, apresentaram problemas ao rastrear grupos de peixes. O maior problema é manter a identidade correta de cada indivíduo ao longo do experimento devido às constantes oclusões que acontecem entre dois ou mais indivíduos. Esse problema é agravado quando os métodos são aplicados em experimentos com mais peixes [33]. Já os métodos de rastreamento que utilizam técnicas 3D, apresentam melhores resultados pois estes exploram mais de uma visão do aquário para um mesmo *frame*. Estes métodos, entretanto, necessitam de um extenso período de calibração para posicionar os espelhos, lâmpadas ou câmeras.

Além disso, dos métodos de rastreamento avaliados na seção 4, nenhum apresentou resultados satisfatórios para manter a identidade dos animais, também, devido a situações de oclusão. Os métodos baseados em intensidades de cor perderam facilmente a identidade, antes mesmo dessas situações. O Filtro de Kalman, por sua vez, foi capaz de manter a referência de um peixe quando outros se aproximaram, somente perdendo-a em casos de oclusão. Pela literatura, observa-se que este método é, de fato, bastante comum em tarefas de rastreamento de objetos, inclusive de peixes. No trabalho de Zhu e Weng [18], por exemplo, o Filtro mantém o rastro de cada peixe nas imagens provenientes da câmera e de seus respectivos reflexos nos espelhos. Já no trabalho de Wu et al. [19], o Filtro mantém o rastro de cada animal nas imagens de ambas as câmeras.

Com base nestas observações, o método desenvolvido nesse trabalho consiste em uma abordagem 2D e tem o objetivo de rastrear cada peixe de um cardume e tratar casos de oclusão através do particionamento de um blob que contenha mais de um peixe. Não está no escopo deste trabalho identificar automaticamente a quantidade de peixes no aquário, logo, para os testes realizados, esta informação é sabida *a priori*.

Para rastrear os peixes do aquário, foi utilizado o Filtro de Kalman. Dado que cada animal tem suas próprias características de movimentação, como aceleração, velocidade e direção, a cada um deles é associada, no primeiro *frame* do vídeo, uma instância do Filtro de Kalman.

A partir deste ponto, quando dois ou mais peixes estiverem sobrepostos na imagem, haverá mais instâncias do Filtro de Kalman do que peixes na imagem do aquário, o que pode fazer o Filtro trocar a referência dos animais envolvidos ou simplesmente perdê-las.

Para evitar este problema, uma técnica de particionamento de blobs é proposta com o objetivo de separar os peixes.

Uma técnica de particionamento de blob similar a utilizada no presente trabalho foi proposta por Kato et al. [16]. No método, são realizadas sucessivas operações morfológicas de erosão até que a região do candidato seja dividida em duas. Após a divisão, cada sub-região passa a representar a posição do peixe mais próximo e a mesma quantidade de operações de dilatação é realizada com o intuito de recuperar a área original.

A técnica de particionamento desenvolvida neste trabalho difere da técnica proposta por Kato et al. [16], principalmente no resultado obtido com as operações morfológicas. Enquanto aqueles autores obtêm a posição de cada peixe após aplicar operações de erosão, neste trabalho a ideia é reconstruir a área de cada peixe, com base na área do candidato. Dessa forma, é possível corrigir a estimativa do Filtro de Kalman com as informações obtidas por essas novas áreas e produzir estimativas mais confiáveis para o próximo *frame*.

5.1 Resumo do Método

O diagrama completo do método desenvolvido está apresentado na Figura 23. Este método pode ser dividido em quatro etapas. Na primeira etapa, Detecção de Blobs e Extração de Características, uma técnica de remoção de fundo é responsável por separar os peixes do restante do cenário. Dessa forma, cada peixe passa a ser representado por um blob na imagem. A partir desses blobs, um conjunto de informações é obtido com o intuito de caracterizar e diferenciar um dado peixe dos demais. A Figura 24 ilustra um exemplo de peixes detectados em uma imagem. A cada blob é associada uma instância do Filtro de Kalman, com o objetivo de utilizar o histórico da posição e da velocidade do animal, para estimar sua nova posição.

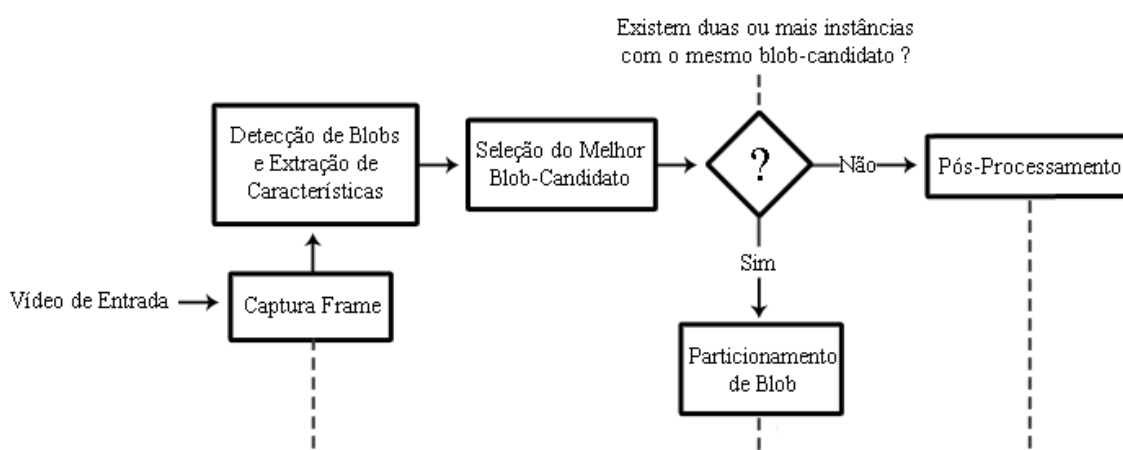


Figura 23 - Diagrama do método proposto.

No *frame* inicial, qualquer blob pode ser associado a qualquer instância do Filtro. Porém, nos *frames* subsequentes, existe apenas um blob que pode ser associado a uma dada instância do Filtro. Por conta disso, a segunda etapa do algoritmo, Seleção do Melhor Blob-Candidato, é executada a partir das propriedades extraídas de cada peixe na etapa anterior. Nesta segunda etapa, são utilizados dois métodos para selecionar qual blob é associado a qual instância do filtro de Kalman. O primeiro é baseado na intersecção de blobs e o segundo é baseado no blob mais próximo.

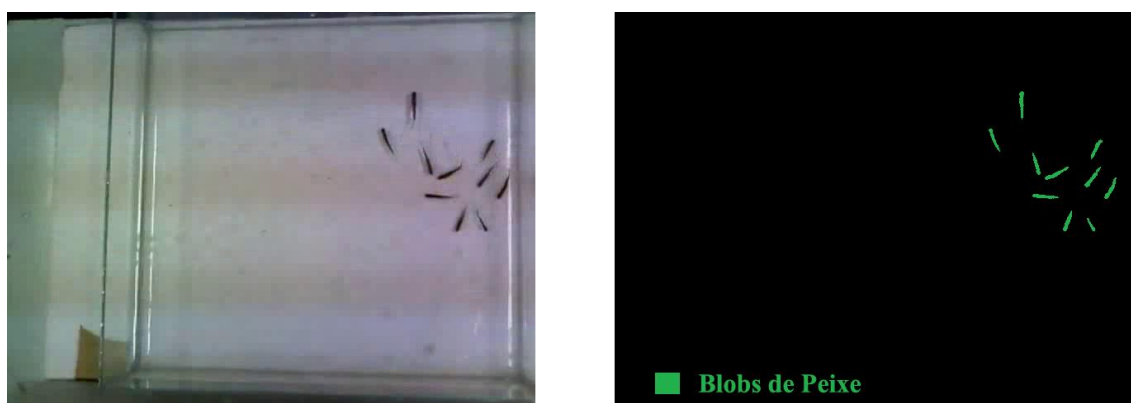


Figura 24 - Exemplo de peixes detectados na etapa 1 do algoritmo.

Sempre que ocorrer oclusão entre peixes, o número de blobs-candidatos será menor que o número de instâncias do filtro, resultando em duas ou mais instâncias apontando para um mesmo candidato, conforme exemplo da Figura 25. Neste caso, a terceira etapa, Particionamento de Blob, é iniciada para particionar o blob-candidato em tantos blobs quantas forem as instâncias envolvidas na oclusão. O particionamento do candidato é realizado de acordo com a área e a orientação do peixe, armazenadas no histórico de cada instância.

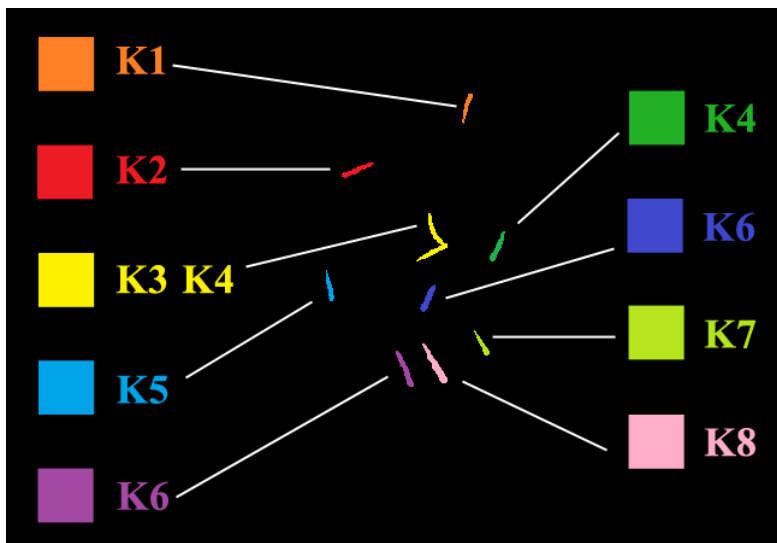


Figura 25 - Exemplo de situação em que há mais instâncias do Filtro de Kalman que blobs-candidatos.

Quando houver apenas uma instância para cada blob-candidato, a etapa de Pós-Processamento é responsável por corrigir a estimativa de cada instância e seus respectivos históricos, com as informações do blob-candidato no *frame* atual.

5.2 Detecção de Blobs e Extração de Características

Os peixes são identificados em cada *frame* utilizando uma técnica de remoção de fundo. Como o cenário de fundo não muda bruscamente durante o experimento, um modelo de fundo foi obtido antes de iniciar o processo de rastreamento e o mesmo modelo foi utilizado até o fim do vídeo. Na Figura 26 pode-se observar um exemplo de um modelo de fundo obtido de um aquário com dimensões 30 x 10 x 30 cm (largura, altura e comprimento).

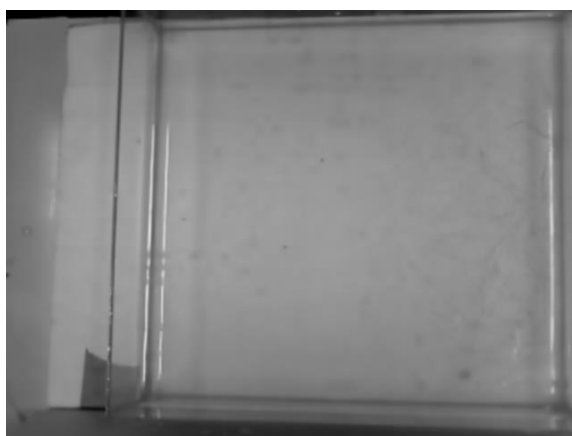


Figura 26 - Modelo de fundo capturado no início do processo de rastreamento.

Ao comparar o modelo de fundo com o *frame* atual, um conjunto de blobs C_t é destacado na imagem. Neste conjunto, t representa o instante de tempo em que os blobs

foram obtidos. Dentre esses blobs estão tanto os blobs dos peixes quanto blobs ruidosos, provenientes de variações na iluminação ou do próprio dispositivo de captura, por exemplo, conforme o exemplo da Figura 27.

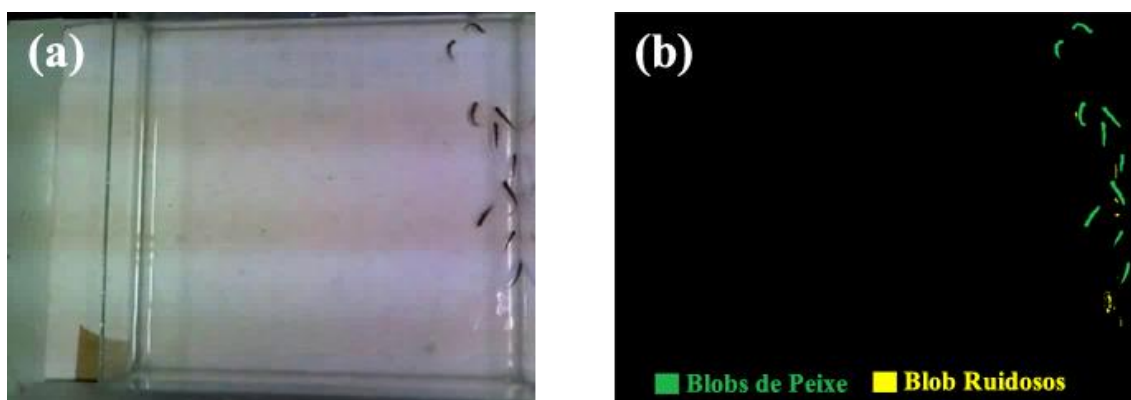


Figura 27 - Resultado da técnica de remoção de fundo.

Com o objetivo de remover os blobs ruidosos do conjunto C_t , o tamanho e a coloração do peixe-zebra são utilizados como critério. Blobs resultantes da reflexão da luz, como pode ser visto na Figura 28, apresentam pixels com intensidades muito altas e, portanto, um processo de limiarização é capaz de remove-los sem afetar os blobs pertencentes aos peixes. Também, são descartados blobs cuja área é menor que um valor definido empiricamente (50 pixels para os vídeos testados), conforme a Figura 29.

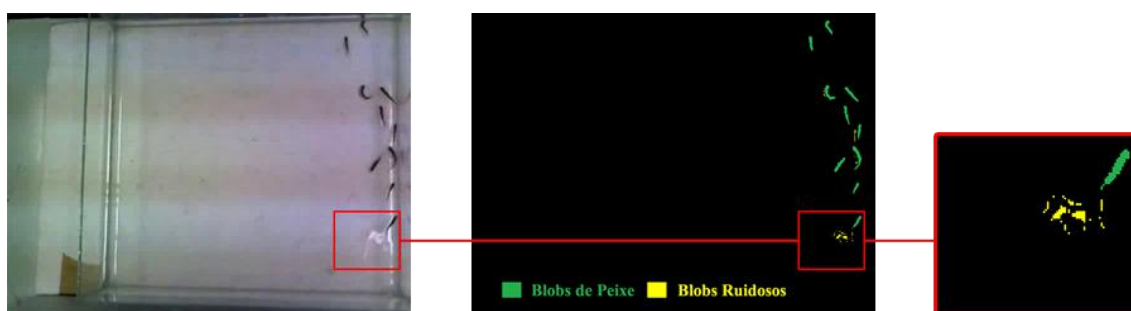


Figura 28 - Exemplo de blobs ruidosos devido a reflexão da luz

Para cada blob C_{it} obtido após o processo de segmentação, em que i representa o índice do blob no conjunto C_t , são extraídas as propriedades de área, mediana geométrica e orientação, que serão utilizadas nas próximas etapas do algoritmo. Estas propriedades são armazenadas no histórico do blob tão logo este seja associado a uma instância do Filtro de Kalman.

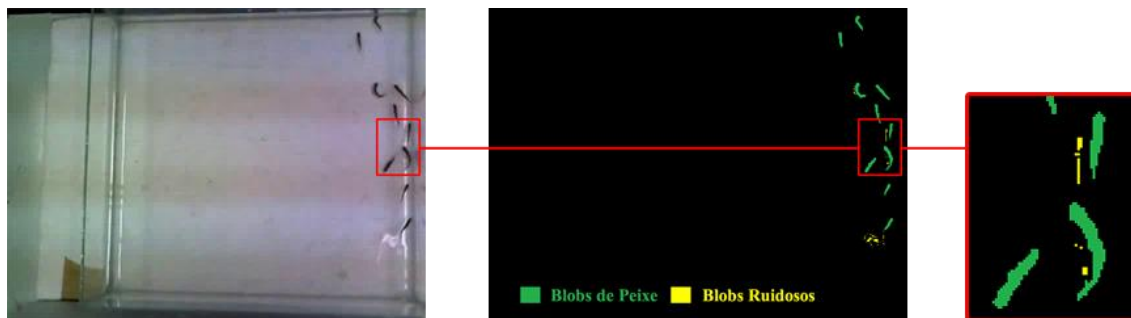


Figura 29 - Exemplo de blobs pequenos

A **área** é definida pela quantidade de pixels contíguos do blob e será usada para restringir a expansão do blob na etapa de Particionamento de Blob. A **mediana geométrica**, que servirá para representar a posição do peixe, é calculada conforme segue. Dado um conjunto de pontos bidimensionais P , que representam um dado blob na imagem segmentada, e sendo n o número de elemento de P , a ideia é encontrar o ponto M , tal que M possua a menor distância para todos os elementos do conjunto P , conforme abaixo.

$$M = \min_{M \in P} \sum_{i=0}^n \|P_i - M\|. \quad (6)$$

Outros trabalhos como de Jianhui et al. [71] e Wang [72] utilizam o Centroide do blob para representar a posição. Esta medida entretanto, por vezes produz posições fora dos limites do blob propriamente dito, conforme os exemplos da Figura 30.

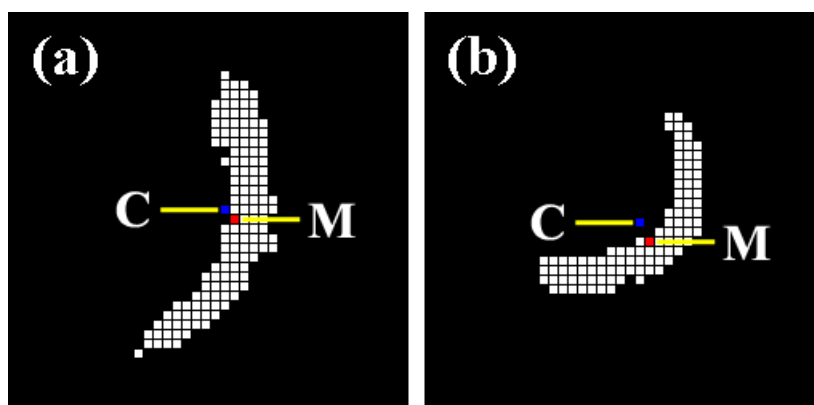


Figura 30 - Comparação entre Centroide e Mediana Geométrica

Para determinar a **orientação** de um peixe, são calculados dois vetores que partem da mediana geométrica em direção aos extremos opostos do blob, Figura 31. Estes pontos são definidos como sendo os dois pontos E_1 e E_2 , tal que a distância entre eles é máxima, conforme segue:

$$\max_{E_1 \in P, E_2 \in P} \|E_1 - E_2\|, \quad (7)$$

na qual P representa o conjunto de pontos bidimensionais que formam o blob.

Após extrair as propriedades de cada blob da imagem, o processo de seleção do melhor blob-candidato para cada instância do Filtro de Kalman é iniciado.

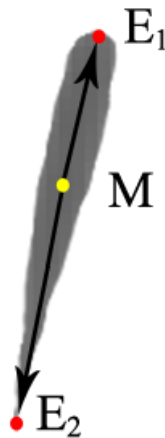


Figura 31 - Mediana Geométrica e Orientação de um Blob

5.3 Seleção do Melhor Blob-Candidato

Deste ponto em diante, os blobs da imagem segmentada no *frame* atual, C_{it} , serão referenciados como blobs-candidatos, para diferenciá-los dos blobs armazenados no histórico de cada instância do Filtro de Kalman, que são chamados de B_{jt} . Os passos seguidos para selecionar o melhor blob-candidato, C_{it} , para cada instância K_j do Filtro de Kalman, são explicados a seguir.

No primeiro *frame* ($t = 0$), cada blob-candidato C_{i0} é associado a uma única instância do Filtro de Kalman. Nos *frames* subsequentes ($t > 0$), cada instância K_j é responsável por produzir uma estimativa X_{jt}^- que será a principal medida utilizada para a seleção do seu melhor candidato. Além dessa medida, o blob $B_{j,t-1}$, associado à instância K_j no *frame* anterior, também é utilizado.

A partir destas informações, o blob $B_{j,t-1}$ (Figura 32a) é transladado de forma que sua mediana geométrica fique alinhada com a estimativa inicial X_{jt}^- , conforme ilustrado na Figura 32b. O resultado dessa translação é o blob inicial B_{jt}^- , destacado na Figura 32c. A seguinte equação resume o processo de obtenção do blob B_{jt}^- ,

$$B_{jt}^- = B_{j,t-1} + (X_{jt}^- - M_{j,t-1}), \quad (8)$$

em que $M_{j,t-1}$ é a mediana geométrica do blob $B_{j,t-1}$.

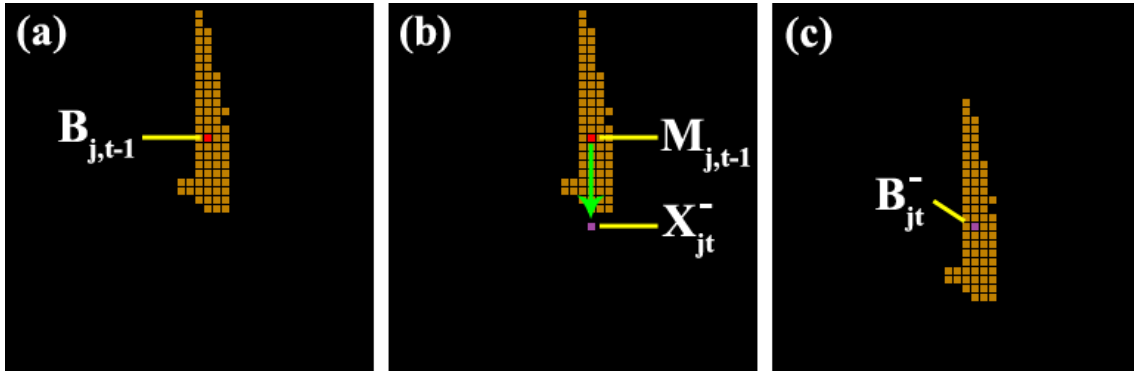


Figura 32 - Processo de translação do blob obtido no *frame* anterior.

Após a obtenção do blob inicial B_{jt}^- , dois métodos são utilizados para a seleção do melhor blob-candidato. No primeiro método, a ideia é escolher um blob-candidato cuja intersecção com o blob B_{jt}^- é máxima. Para este propósito, avalia-se a intersecção do blob B_{jt}^- com cada blob-candidato C_{it} , conforme a equação:

$$C_{it} = \max_{C_{it} \in C_t} (B_{jt}^- \cap C_{it}). \quad (9)$$

A Figura 33 ilustra um exemplo em que o método de intersecção é utilizado para selecionar o melhor blob-candidato. Neste exemplo, existem dois candidatos, identificados como C_{1t} e C_{2t} na Figura 33a. A partir do blob $B_{j,t-1}$, Figura 33b, realiza-se a translação descrita na equação 8 (Figura 33c), obtendo-se o blob inicial B_{jt}^- (Figura 33d).

Como existem dois candidatos, determinam-se as intersecções entre B_{jt}^- e C_{1t} , e entre B_{jt}^- e C_{2t} (Figura 33e), concluindo-se que a intersecção com C_{1t} apresenta uma área de sobreposição superior à área de C_{2t} . Neste caso, o blob C_{1t} é selecionado como blob-candidato.

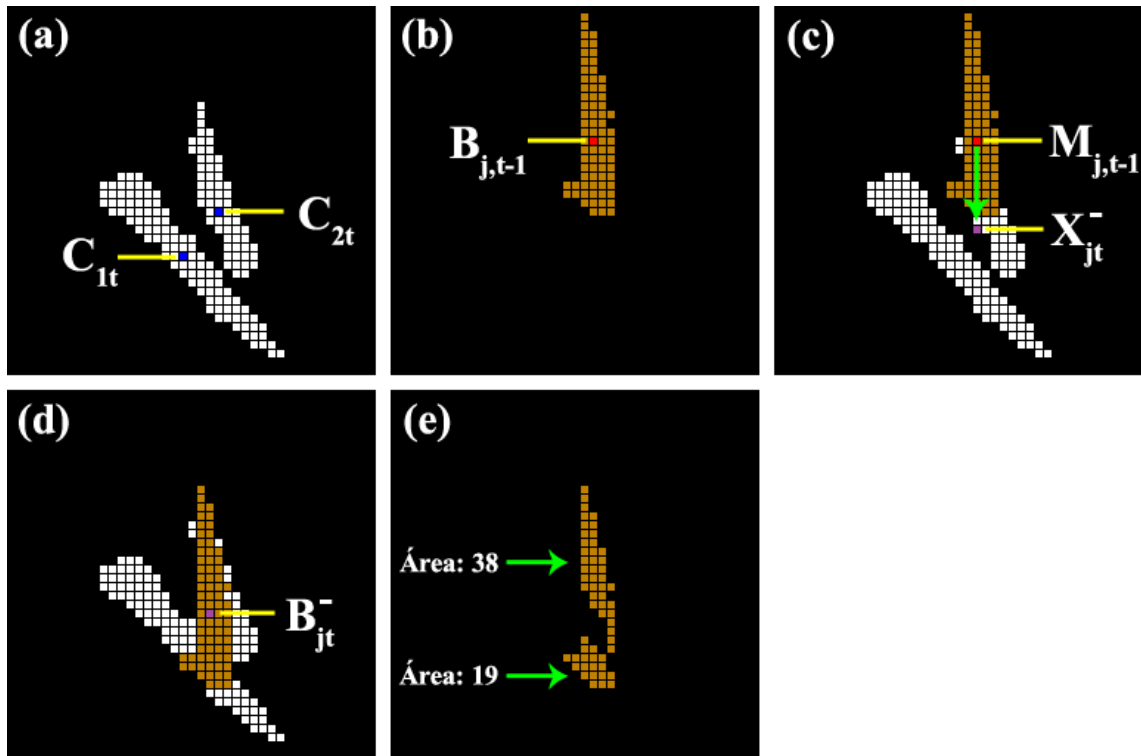


Figura 33 - Processo de seleção de blobs realizado através do método de intersecção.

Para os casos em que há uma intersecção não-vazia, o blob-candidato C_{it} é atribuído para sua respectiva instância K_j do Filtro de Kalman. Entretanto, nem a equação 9, nem a estimativa X_{jt}^- garantem que haverá um blob-candidato C_{it} que interseccione B_{jt}^- , como no exemplo da Figura 34. Nestes casos, o blob-candidato é dado pelo segundo método, que consiste em selecionar o candidato cuja mediana geométrica, N_{it} , estiver mais próxima da estimativa X_{it}^- . A equação

$$C_{it} = \min_{i \geq 0, i < n} |N_{it} - X_{jt}^-| \quad (10)$$

descreve este segundo método, em que i é o índice do blob no conjunto C_t e n é o tamanho desse conjunto. A Figura 35 ilustra um exemplo similar ao anterior, porém, com o uso do método baseado no blob mais próximo. Neste exemplo, é possível observar que o método baseado na distância, seleciona o blob mais próximo de B_{jt}^- , que, neste caso, corresponde ao blob C_{1t} (Figura 35f).

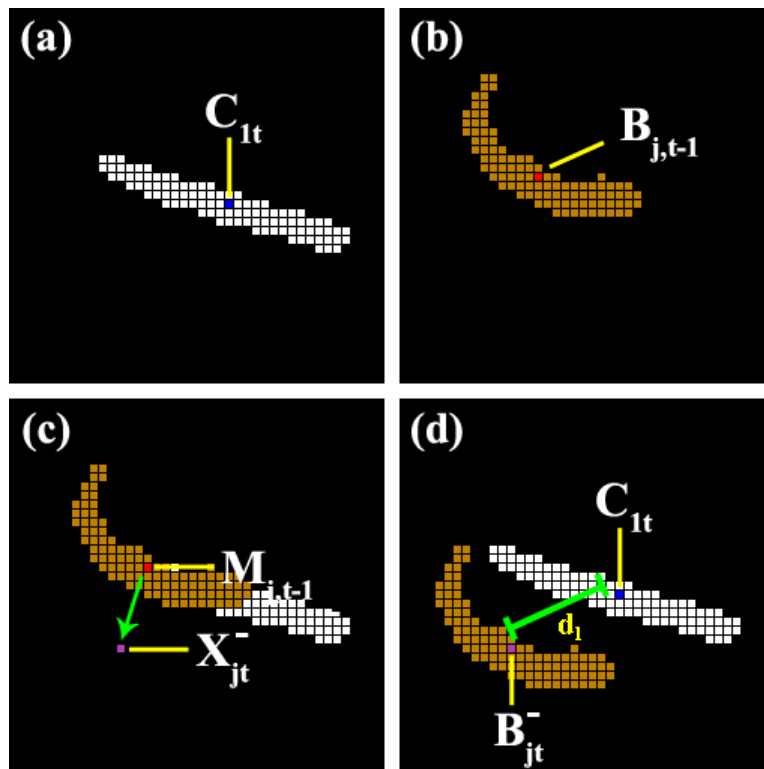


Figura 34 - Processo de seleção de blobs com base no blob mais próximo.

Ao final desta etapa, cada instância K_j terá um blob C_{it} associado a ela. Isto entretanto, não garante que todas as instâncias estejam associadas a um blob diferente, pois pode ter ocorrido alguma sobreposição entre os peixes, como no exemplo da Figura 35. Neste caso, a etapa de Particionamento de Blob é iniciada.

5.4 Particionamento de Blob

Em resumo, o objetivo do processo de particionamento é dividir um blob que tenha sido selecionados para duas ou mais instâncias do Filtro de Kalman (Figura 35a-c), a fim de obter um blob diferente para cada uma. Neste trabalho, o particionamento é realizado através de sucessivas operações morfológicas de dilatação que aumentam a área de ocupação de cada blob a partir de uma região inicial, até que a área histórica desses blobs seja alcançada.

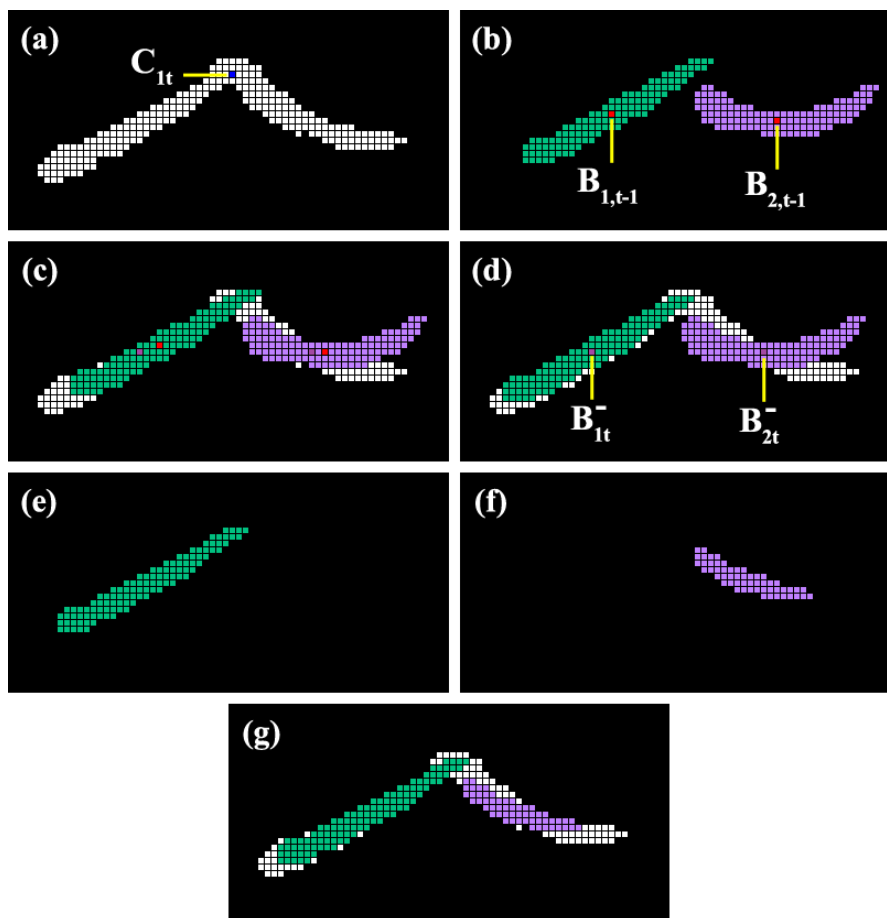


Figura 35 - Exemplo de dois Filtros de Kalman apontando para um o mesmo blob-candidato.

O primeiro passo da etapa de particionamento consiste em definir a região inicial de cada blob. Os mesmos passos de translação e intersecção realizados para seleccionar o melhor blob-candidato, são usados para definir esta região, conforme as imagens (c), (d), (e) e (f) da Figura 35. O que interessa nesta etapa do algoritmo é apenas a região B_{jt}^{\wedge} em que os blobs B_{jt}^{-} e C_{it} se sobrepõem, conforme a equação:

$$B_{jt}^{\wedge} = B_{jt}^{-} \cap C_{it}. \quad (11)$$

Uma vez obtido o blob B_{jt}^{\wedge} , inicia-se um processo de dilatação morfológica [73] da área deste blob. Para esta operação poderia ser escolhido um elemento estruturante fixo, como na abordagem de [16]. Entretanto, para objetos alongados, deformáveis e que podem assumir orientações diversas, como os peixes-zebra, esta abordagem pode não ser interessante, como se pode ver na Figura 36. No exemplo, a ideia é obter o peixe ilustrado em (a), a partir da região inicial destacada em (b). Para tanto, os elementos estruturantes (c) e (d) são aplicados na região inicial e os resultados são ilustrados, respectivamente, nas

imagens (e) e (f). É possível perceber que em ambos os casos, esses elementos estruturantes distorcem a forma original do animal.

De modo a preservar a geometria do blob, o processo de expansão é realizado com base em dois critérios. O primeiro consiste em restringir a direção de expansão do blob de acordo com os seus vetores de orientação, calculados na seção 5.2. Esta orientação irá definir, dinamicamente, qual a forma do elemento estruturante a ser usado na dilatação. Essa restrição pretende que a forma do peixe no *frame* atual, se mantenha coerente com a sua forma obtida no *frame* anterior. Embora o peixe possa mudar sua direção ao longo do percurso, essa mudança não chega a ser significativa entre dois *frames* consecutivos. O segundo critério consiste em manter a área desse blob próxima da área que esse animal apresenta ao longo do experimento.

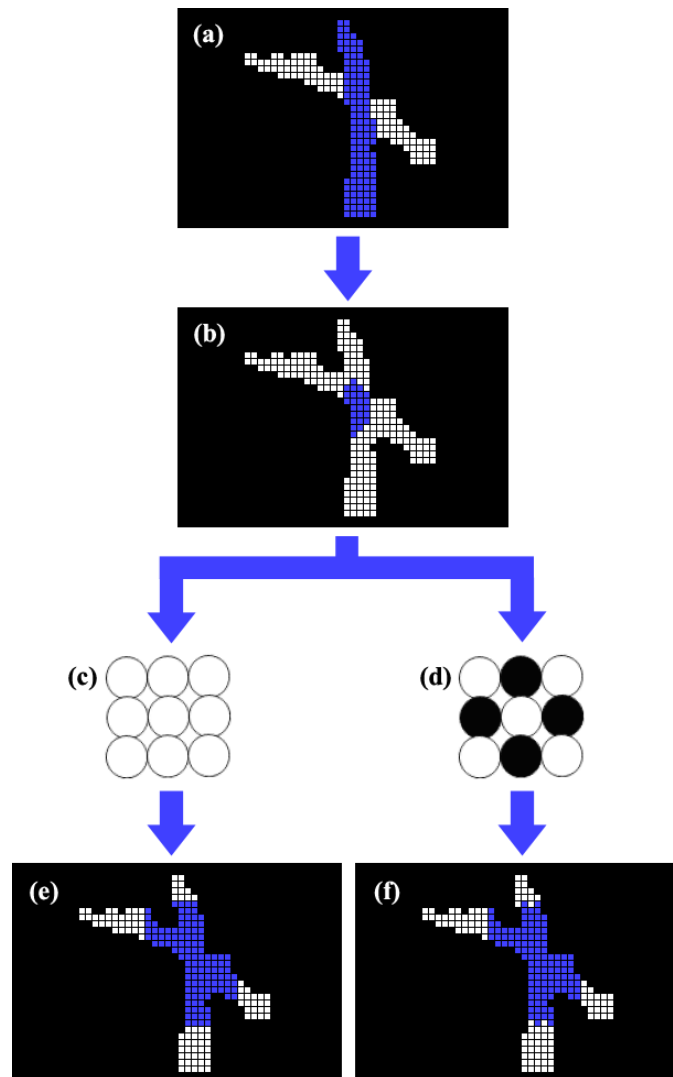


Figura 36 - Exemplo de expansão de um blob de um peixe-zebra com diferentes elementos estruturantes.

A escolha do elemento estruturante é realizada a partir de um ciclo trigonométrico dividido em oito regiões igualmente espaçadas em intervalos de 45° , conforme a Figura 37. Cada região do ciclo define uma orientação possível para um peixe e, conseqüentemente, possui uma máscara associada. Como um blob possui duas orientações, são usadas duas máscaras, $Mask_1$ e $Mask_2$, para realizar sua expansão.

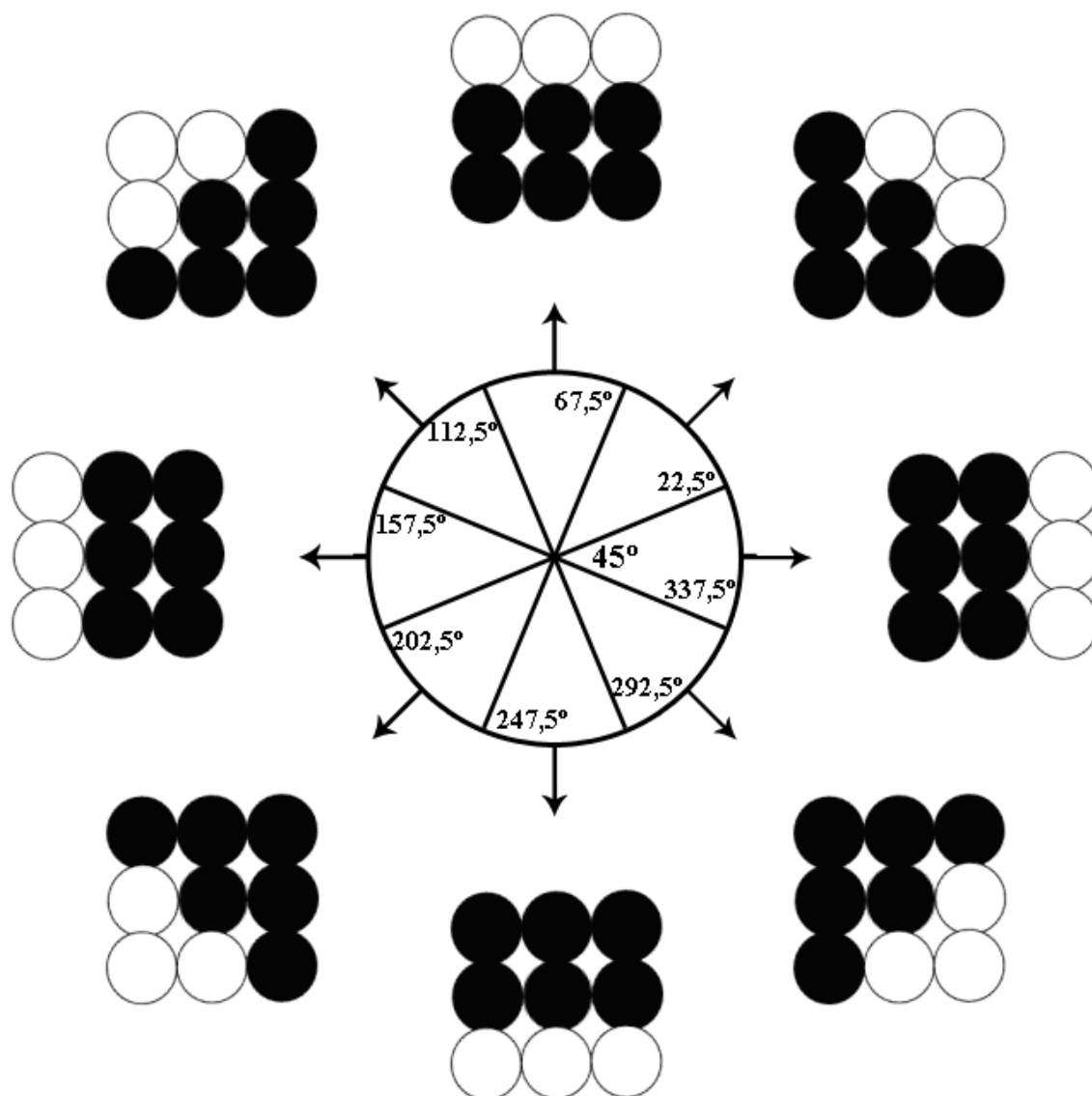


Figura 37 - Ciclo trigonométrico utilizado para definir o elemento estruturante de um blob.

Na Figura 38, são ilustrados alguns exemplos de blobs com seus respectivos elemento estruturantes para a dilatação. Em cada exemplo, o ciclo trigonométrico é posicionado sobre a mediana geométrica do blob e, então, avalia-se em quais direções os vetores orientação apontam.

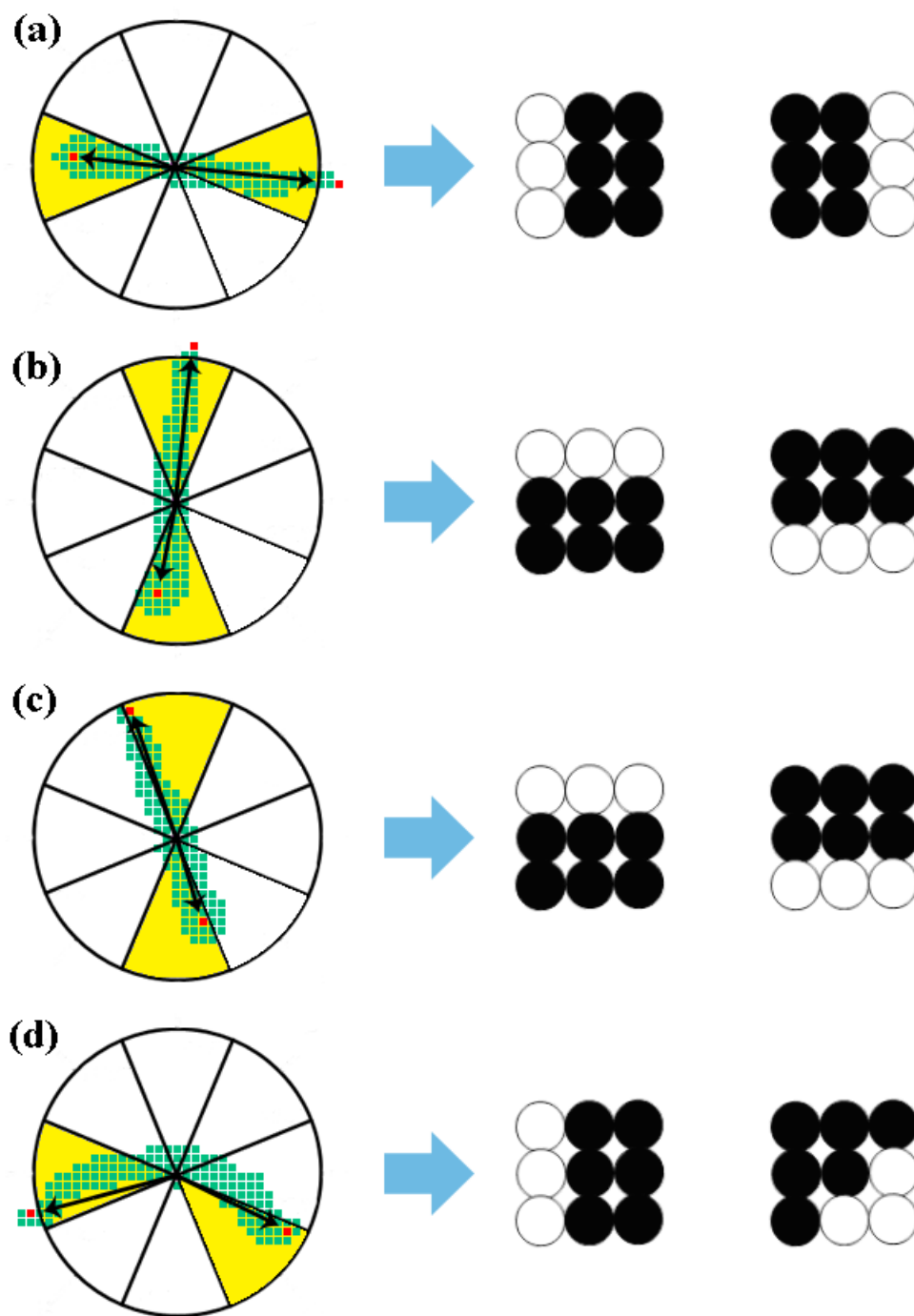


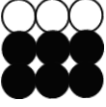
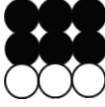
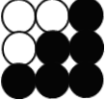
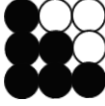
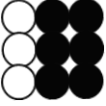
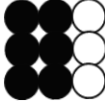

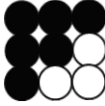
Figura 38 - Exemplos do método de seleção de máscaras de expansão.

Após a seleção dos elementos estruturantes, o processo de expansão é iniciado para cada um dos blobs B_{jt}^{\wedge} .

Inicialmente, cada pixel do blob é analisado para avaliar qual das duas máscaras deve ser aplicada a ele. A decisão é baseada na relação entre a posição do pixel e a posição da mediana geométrica. Na Tabela 3 são apresentadas as condições para a aplicação de

cada uma das possíveis máscaras. Ressalta-se que apenas as duas máscaras já selecionadas passarão pelo processo de avaliação.

Tabela 3 - Condições para expansão do pixel

Máscara	Condição	Máscara	Condição
	$P_y < M_y$		$P_y > M_y$
	$P_x < M_x \text{ e } P_y < M_y$		$P_x > M_x \text{ e } P_y < M_y$
	$P_x < M_x$		$P_x > M_y$
	$P_x < M_x \text{ e } P_y > M_y$		$P_x > M_x \text{ e } P_y > M_y$

(P_x, P_y) : Pixel avaliado; (M_x, M_y) : Mediana Geométrica.

No exemplo da Figura 38a, as máscaras $Mask_1$ e $Mask_2$ apontam, respectivamente, para a esquerda e para a direita. Dessa forma, os pixels do blob que estão à esquerda de sua mediana geométrica, serão expandidos por $Mask_1$, enquanto que os pixels que estão à direita serão expandidos por $Mask_2$.

Após a expansão, caso o processo gere algum pixel fora do blob-candidato C_{it} , sua forma é corrigida com o intuito de manter a coerência com a geometria do candidato C_{it} , utilizando a equação

$$\hat{B}_{jt} = \hat{B}_{jt} \cap C_{it}. \quad (12)$$

Este processo se repete até que a área histórica do blob \hat{B}_{jt} seja alcançada. A Figura 39 ilustra uma iteração do processo de expansão de um blob. Neste exemplo, o blob-candidato, é identificado como C_{1t} (Figura 39a) e o blob associado à uma instância do Filtro de Kalman no *frame* anterior, é identificado como $B_{2,t-1}$ (Figura 39b). A partir de $B_{2,t-1}$, define-se os elementos estruturantes que serão utilizados para a dilatação do blob (Figura 39c). Neste ponto, a mediana geométrica, M_{2t}^{\wedge} , do blob B_{2t}^{\wedge} (Figura 39d), é utilizada para selecionar os pixels do blob de acordo com a máscara de expansão associada a eles

(imagens (e) e (f) da Figura 39). No exemplo, os pixels a esquerda de M_{2t}^{\wedge} representam a parte p_1 , enquanto os pixels da direita, representam a parte p_2 . É possível observar que p_1 , é dilatada somente para a esquerda (Figura 39g), enquanto p_2 , somente para a direita (Figura 39h). Após a dilatação, o blob B_{2t}^{\wedge} apresenta diversos pixels fora dos limites do candidato, que são removidos com o uso da equação 12, como pode ser visto nas imagens (i) e (j) da Figura 39.

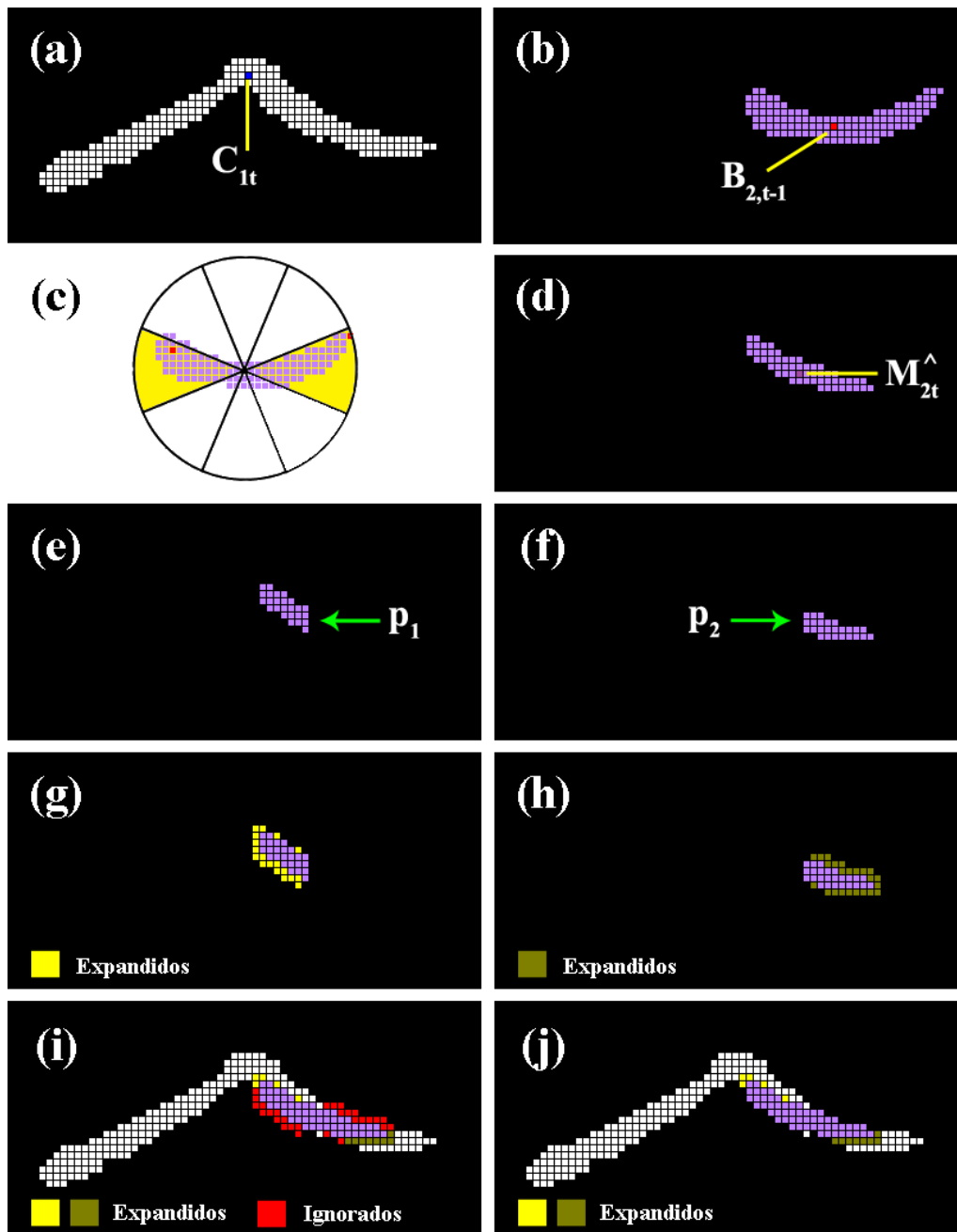


Figura 39 - Passo a passo do processo de expansão de um blob.

A Figura 40 ilustra o restante das iterações do processo de expansão do blob B_{2t}^{\wedge} . Cada linha da figura representa uma iteração diferente, em que as imagens da esquerda

ilustram os pixels expandidos antes de aplicar a equação 12, e as imagens da direita ilustram a expansão após o uso da equação.

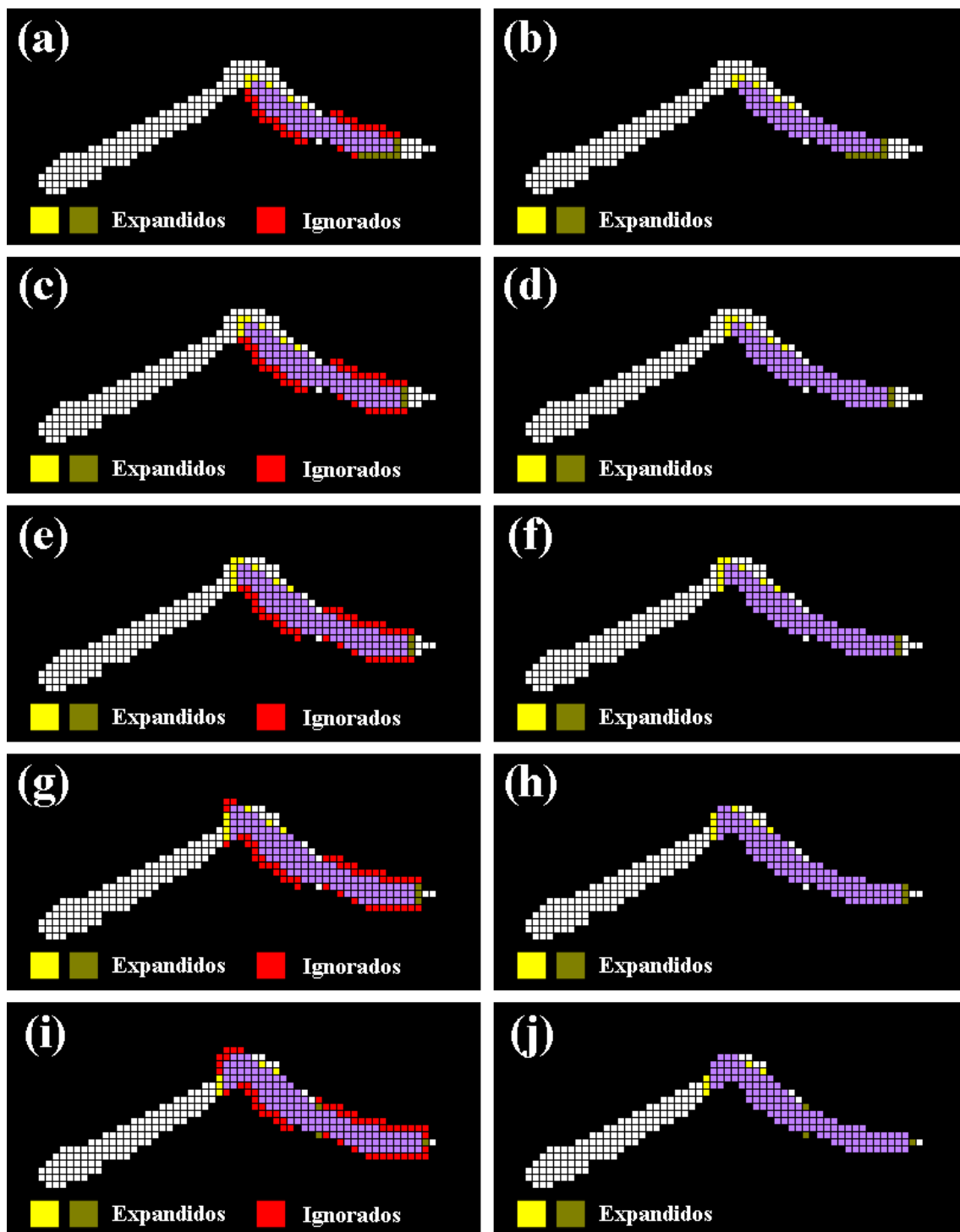


Figura 40 - Sequência de iterações realizadas por um blob até atingir sua área histórica.

A Figura 41 ilustra, em linhas gerais, o cenário completo do processo de particionamento realizado com o blob da Figura 40, em que ambos os blobs são expandidos. A Figura 41a ilustra o blob-candidato C_{1t} e a Figura 41b ilustra os blobs $B_{1,t-1}$

e $B_{2,t-1}$. As imagens (c), (d), (e) e (f) da Figura 41 ilustram o processo de translação e obtenção da região inicial de cada blob B_{jt}^{\wedge} . Logo após, o processo de expansão é iniciado para cada blob até que a área histórica do blob seja alcançada. O resultado final, pode ser visto na Figura 41g-i.

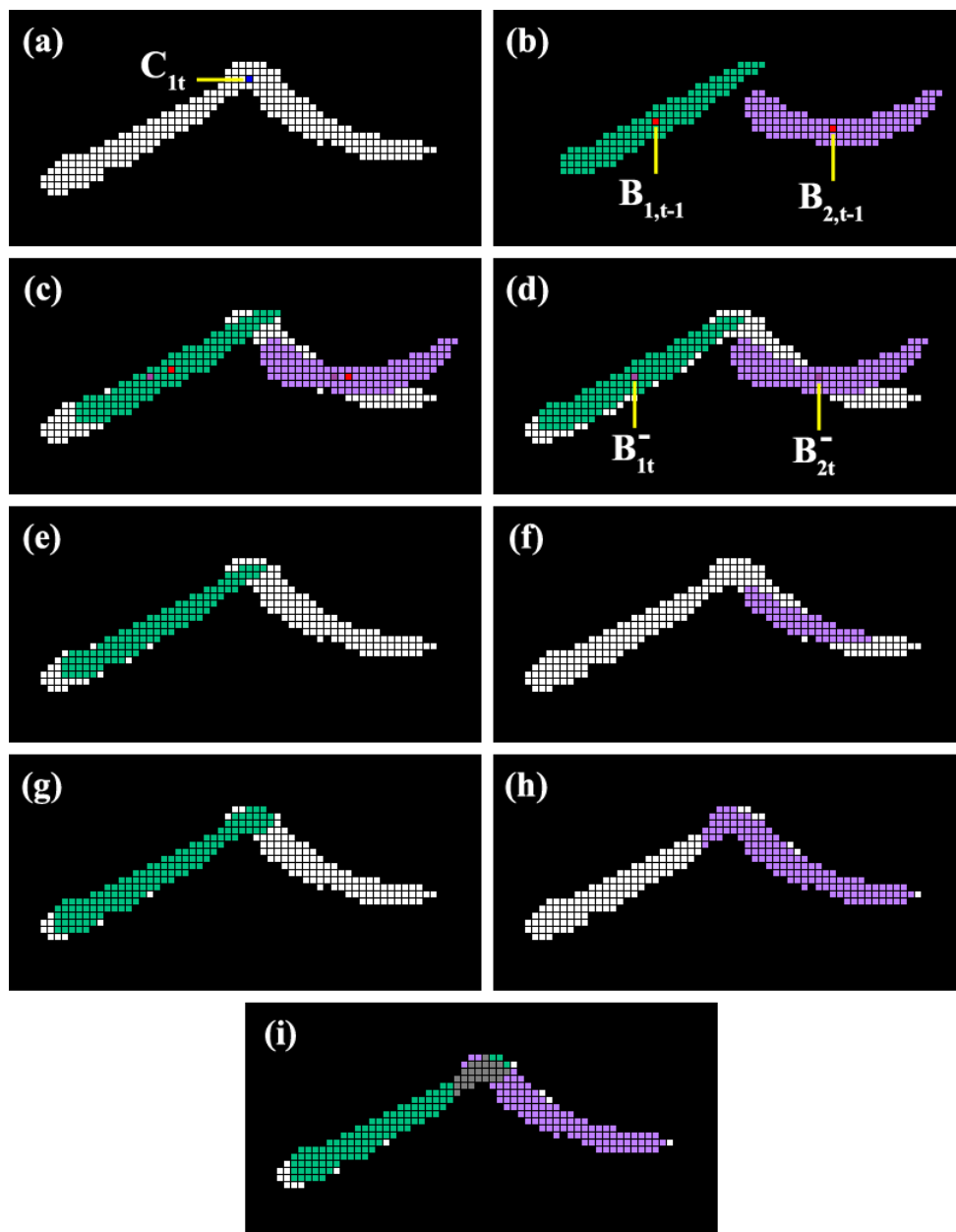


Figura 41 - Exemplo de resultado obtido após a realização do processo de expansão em dois blobs.

5.5 Pós-Processamento

Na última etapa do algoritmo, cada instância K_j já tem um blob B_{jt} distinto, proveniente da etapa de Seleção do Melhor Blob-Candidato, C_{it} , ou da etapa de Particionamento de Blobs, B_{jt}^{\wedge} .

O objetivo aqui é atualizar todas as informações necessárias para o próximo *frame*. Primeiro, a estimativa X_{jt}^- é corrigida com base na mediana geométrica de B_{jt} . Segundo, o conjunto de propriedades mencionadas na seção 5.2 é armazenada no histórico de sua respectiva instância K_j .

6 RESULTADOS

Com o intuito de avaliar o método de rastreamento proposto, as duas etapas principais foram testadas a partir de um *ground truth* obtido manualmente. Primeiro, a taxa de acerto da etapa de seleção do melhor blob-candidato é avaliada e comparada com diferentes métodos existentes na literatura. No segundo teste, apenas a performance da etapa de Particionamento de Blob é avaliada.

A avaliação foi realizada em trechos de dois vídeos contendo um aquário com 10 peixes da espécie peixezebra cada. Estes mesmos vídeos foram utilizados nos testes dos algoritmos da seção 4. A configuração do cenário em que estes vídeos foram capturados está retratada na Figura 42. Os vídeos foram obtidos por uma câmera iLook 1321 V2 (Figura 43) posicionada 30cm acima do aquário, distância suficiente para o aquário inteiro ser capturado. Cada vídeo foi capturado à uma taxa de 30 *frames* por segundo com resolução espacial de 640x480.

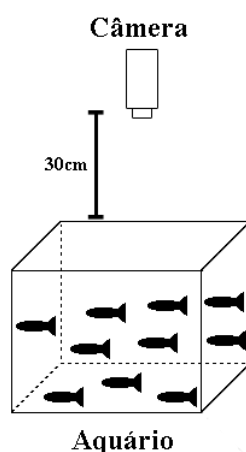


Figura 42 - Ilustração do cenário onde foram obtidos os vídeos.



Figura 43 - Câmera iLook 1321 V2 utilizada para capturar os vídeos.

A quantidade de *frames* e ocorrências de oclusão dos vídeos testados pode ser visualizada na Tabela 4.

Tabela 4 - Informações sobre os vídeos A e B testados.

	A	B
Quantidade de <i>frames</i>	1945	1940
Ocorrências de oclusão	800	880

Para operar, o algoritmo recebe a quantidade total de animais no aquário e a área mínima que o blob de um peixe pode apresentar nas imagens. Essas informações são necessárias para inicializar a quantidade correta de instâncias do Filtro de Kalman e também para filtrar blobs ruidosos.

Inicialmente, o algoritmo percorre o vídeo até encontrar o primeiro *frame* em que todos os animais estão separados e inicializa uma instância do filtro para cada animal. Após este *frame* inicial, o vídeo é analisado por 1 minuto.

6.1 Avaliação do método de seleção de candidatos

Para escolher o melhor blob-candidato para uma dada instância do Filtro de Kalman, o método de seleção de candidatos (seção 5.3) utiliza ambas as técnicas de intersecção entre blobs e do blob mais próximo pela mediana geométrica. Por esta razão, no texto a seguir, este método recebe o nome de INT+MG.

Com o objetivo de avaliar a performance do método proposto, este foi comparado com as seguintes abordagens:

- Centroide(C): Nesta abordagem, a ideia é selecionar um candidato com base no centroide do blob, como nos trabalhos de Jianhui et al. [71] e Wang et al. [72];
- Mediana Geométrica(MG): Aqui a ideia é utilizar apenas a Mediana Geométrica para selecionar o candidato;
- Intersecção(INT): Nesta abordagem utiliza-se apenas o método de intersecção apresentado na seção 5.3 para selecionar o candidato;
- Intersecção + Centroide(INT+C): A ideia é tentar selecionar o candidato com base no método INT e, caso este falhe, o método C é usado para selecionar o candidato mais próximo.

Para avaliar o desempenho dos métodos, estes foram comparados com um *ground truth*. Neste *ground truth* cada animal presente no vídeo foi identificado em todos os *frames*.

Para evitar que um erro de um algoritmo se propague por todos os quadros após a ocorrência de um erro, nos casos em que um candidato incorreto for selecionado, o candidato correto (a partir do *ground truth*) é atribuído à respectiva instância do Filtro de Kalman de forma que a iteração seguinte não seja prejudicada. Os resultados desta comparação podem ser vistos na Tabela 5, que mostra a quantidade de vezes que cada método identificou incorretamente um peixe.

Tabela 5 - Quantidade total de erros de cada método em comparação com o *ground truth*.

	C	MG	INT	INT+C	INT+MG
Erros	36	45	25	14	7

Os métodos C e MG apresentaram os piores resultados dentre os cinco métodos avaliados. O motivo principal desses resultados está ligado a situações de oclusão parcial, em que um novo animal se aproxima dos animais envolvidos em oclusão, como pode ser visto na Figura 44. Nestes casos, a referência do novo animal pode ser atribuída incorretamente.

O método INT, por outro lado, não sofre desse problema, uma vez que apenas a correlação entre o blob-candidato e o blob associado a uma instância do filtro no *frame* anterior, é avaliada. A maior parte das atribuições incorretas do método INT estão relacionadas a mudanças bruscas na direção de movimento dos peixes. Nestes casos, o blob-candidato pode não estar perto o bastante da estimativa produzida pelo Filtro de Kalman e, conseqüentemente, não é capaz de selecionar um candidato, como pode ser visto na Figura 45. Os métodos compostos (INT+C e INT+MG) apresentaram os melhores resultados, pois quando um método falha, uma segunda opção é utilizada para selecionar o blob-candidato.

Além de apresentar o melhor desempenho dentre todos os métodos, o método proposto foi capaz de corrigir boa parte dos casos em que os outros métodos falham, sem introduzir uma quantidade significativa de erros. Do total de associações incorretas feitas pelos cinco métodos (C, MG, INT, INT+C, INT+MG), o método proposto foi capaz de recuperar pelo menos 64,3% delas, como pode ser visto na Tabela 6. Considerando o número total de *frames* avaliados (3.885) e a quantidade de peixes em cada vídeo (10), o método proposto foi capaz de selecionar o melhor blob-candidato em 99,82% dos casos.

Tabela 6 - Performance do método proposto em comparação com os métodos de seleção de candidatos avaliados.

	C		MG		INT		INT+C	
	QA	QE	QA	QE	QA	QE	NH	QE
INT+MG	31	2	39	1	19	1	9	3
	86,1%	5,5%	86,6%	2,2%	76,0%	4,0%	64,3%	24,4%

QA: Quantidade de acertos do método INT+MG em relação a um determinado método. QE significa a quantidade de erros introduzidos pelo método INT+MG.

A Figura 44 ilustra um exemplo em que o método MG falha, enquanto que o método INT atribui corretamente o blob-candidato. Neste exemplo, os blobs-candidatos são identificados como C_{1t} e C_{2t} (Figura 44a) enquanto os blobs relacionados às instâncias de Kalman são identificados como $B_{1,t-1}$, $B_{2,t-1}$ e $B_{3,t-1}$ (Figura 44b). Na Figura 44c, os cinco blobs são apresentados no mesmo cenário. Na Figura 44d, cada blob $B_{j,t-1}$ é transladado com base em sua respectiva estimativa X_{jt}^- , logo, C_{1t} é o candidato escolhido para B_{1t}^- e C_{2t} é o candidato associado à B_{3t}^- .

O problema acontece com B_{2t}^- , pois a distância d_1 , entre B_{2t}^- e C_{1t} é menor que a distância d_2 , entre B_{2t}^- e C_{2t} . O blob B_{2t}^- será, então, associado à um candidato incorreto.

A seleção de candidatos com o método INT, por outro lado, produz um resultado correto, pois cada blob intersecciona o seu respectivo blob-candidato, conforme a Figura 44f.

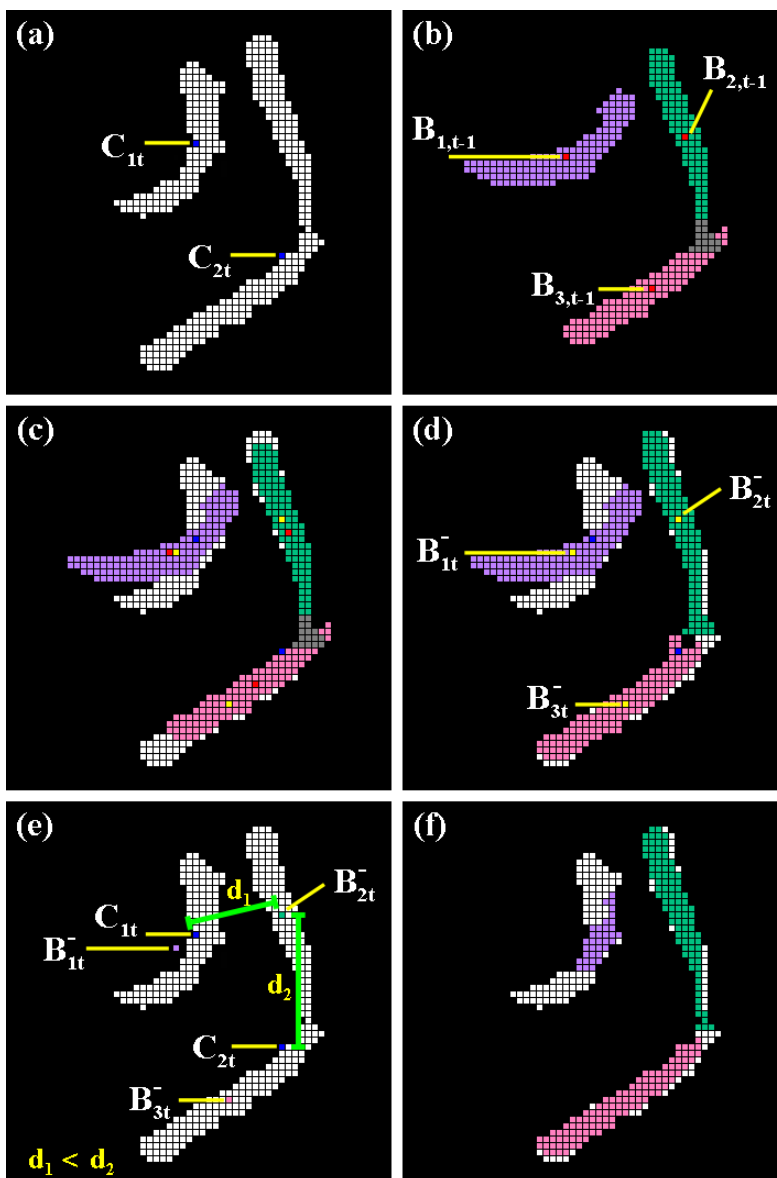


Figura 44 - Exemplo de atribuição falha com o método MG e correta com o método INT.

A Figura 45 ilustra um exemplo em que o método INT falha, porém o método MG acerta. Neste exemplo, percebe-se que o blob B_{2t}^- não intersecciona um blob-candidato (Figura 45e) logo, este blob não será associado com nenhum candidato. O método MG, por outro lado, seleciona o candidato C_{2t} , pois a distância d_2 , entre B_{2t}^- e C_{2t} , é menor que as distâncias d_1 e d_3 , entre os blobs B_{2t}^- e C_{1t} e B_{2t}^- e C_{3t} , respectivamente. A Figura 45f ilustra esse resultado.

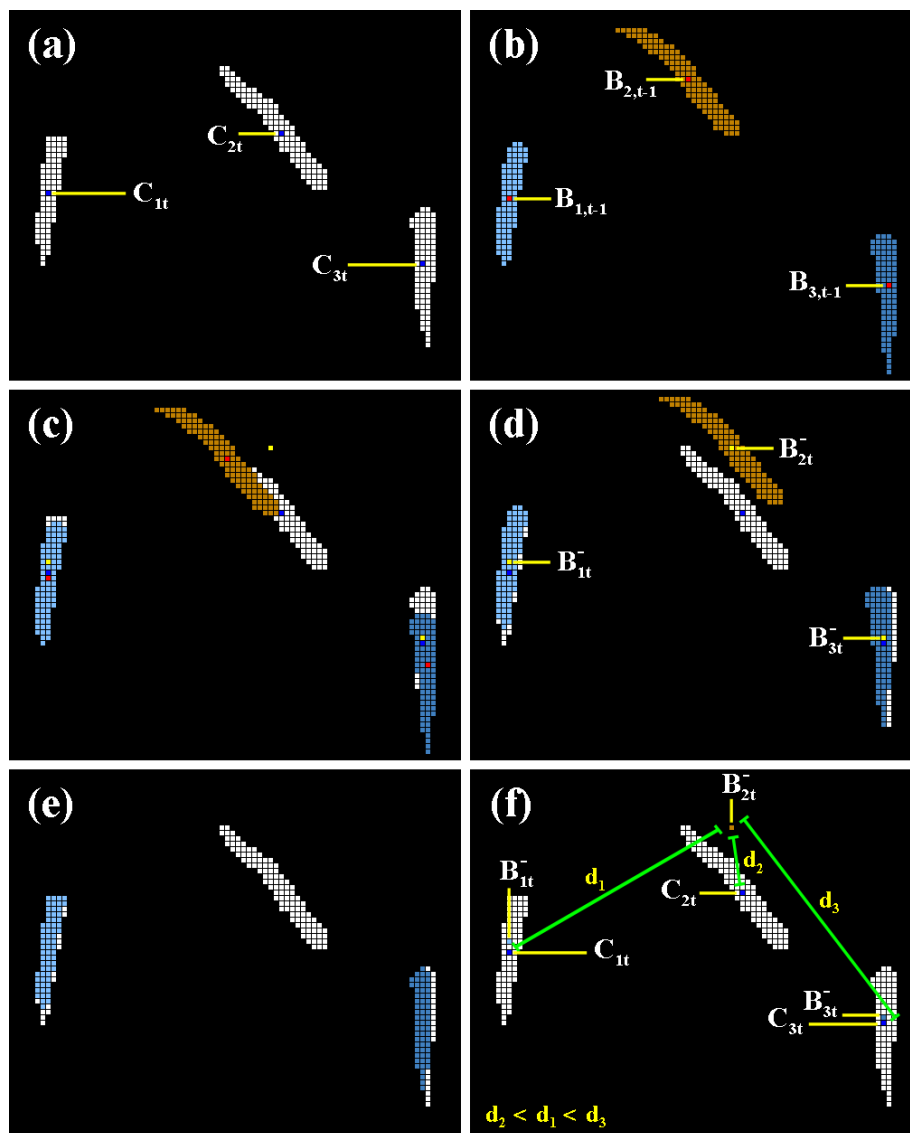


Figura 45 - Exemplo de atribuição falha com o método INT e correta com o método MG.

6.2 Avaliação do método de particionamento

Com o objetivo de testar o algoritmo de particionamento de blobs, cada ocorrência de oclusão foi comparada pixel a pixel com um *ground truth* obtido manualmente.

Um particionamento foi considerado incorreto se pelo menos 20% de sua área não coincidir com o *ground truth* ou se o algoritmo perder a referência de pelo menos um peixe durante o processo de particionamento. Para garantir que os erros provenientes de um particionamento falho não afetem a próxima iteração, em casos de erro, o histórico é atualizado artificialmente com as informações do *ground truth*.

Considerando os resultados obtidos pelo método de particionamento proposto, apresentados na Tabela 7, é possível observar que este é capaz de reconstruir a área

correta de cada indivíduo em 98,04% das ocorrências quando 2, 3 ou 4 peixes estão sobrepostos.

Tabela 7 - Resultados obtidos pelo método proposto em comparação com o *ground truth*.

Peixes em oclusão	Total	Acertos	Erros	Taxa de acerto
2	1585	1560	25	98,4%
3	75	71	4	94,7%
4	14	13	1	92,9%
5	6	3	3	50,0%
Total	1680	1647	33	98,0%

O desempenho é comprometido em algumas situações em que dois ou mais animais estão em oclusão e um deles muda de direção. Nestes casos, as máscaras selecionadas podem prejudicar a expansão do blob, principalmente quando a orientação do segundo é similar à do primeiro, como pode ser visto no exemplo da Figura 46.

Além disso, quanto maior a quantidade de peixes envolvidos em uma oclusão, maior é a chance que este erro ocorra. Ainda na Tabela 7, é possível observar que a performance do algoritmo diminui na medida em que mais peixes estão envolvidos em oclusão. Situações em que 5 peixes estão sobrepostos, por exemplo, apresentam uma taxa de acerto de 50%, embora estas ocorrências correspondem a menos de 1% do total de casos de oclusão encontrados nos vídeos.

A Figura 46 ilustra um exemplo em que um blob envolvido em oclusão expande na direção errada. Neste exemplo, existe apenas um candidato C_{1t} , Figura 46a, e dois blobs de instância, $B_{1,t-1}$ e $B_{2,t-1}$, Figura 46b. Nas Figura 46c-d são ilustrados os blobs antes e depois da etapa de translação, respectivamente. É possível observar que o blob B_{1t}^- apresenta uma orientação levemente inclinada se comparado com a região do candidato C_{1t} que ele deverá ocupar. Após definir as máscaras de B_{1t}^- com base em suas orientações, Figura 46e, sua expansão é iniciada e são necessárias 42 iterações para que sua área histórica seja alcançada, Figuras 46f-j. Percebe-se que, a partir da iteração 20, o blob começa a expandir em direção à B_{2t}^- , distorcendo sua forma final. Na Figura 46j, é ilustrado o resultado obtido ao final da última iteração de B_{1t}^- .

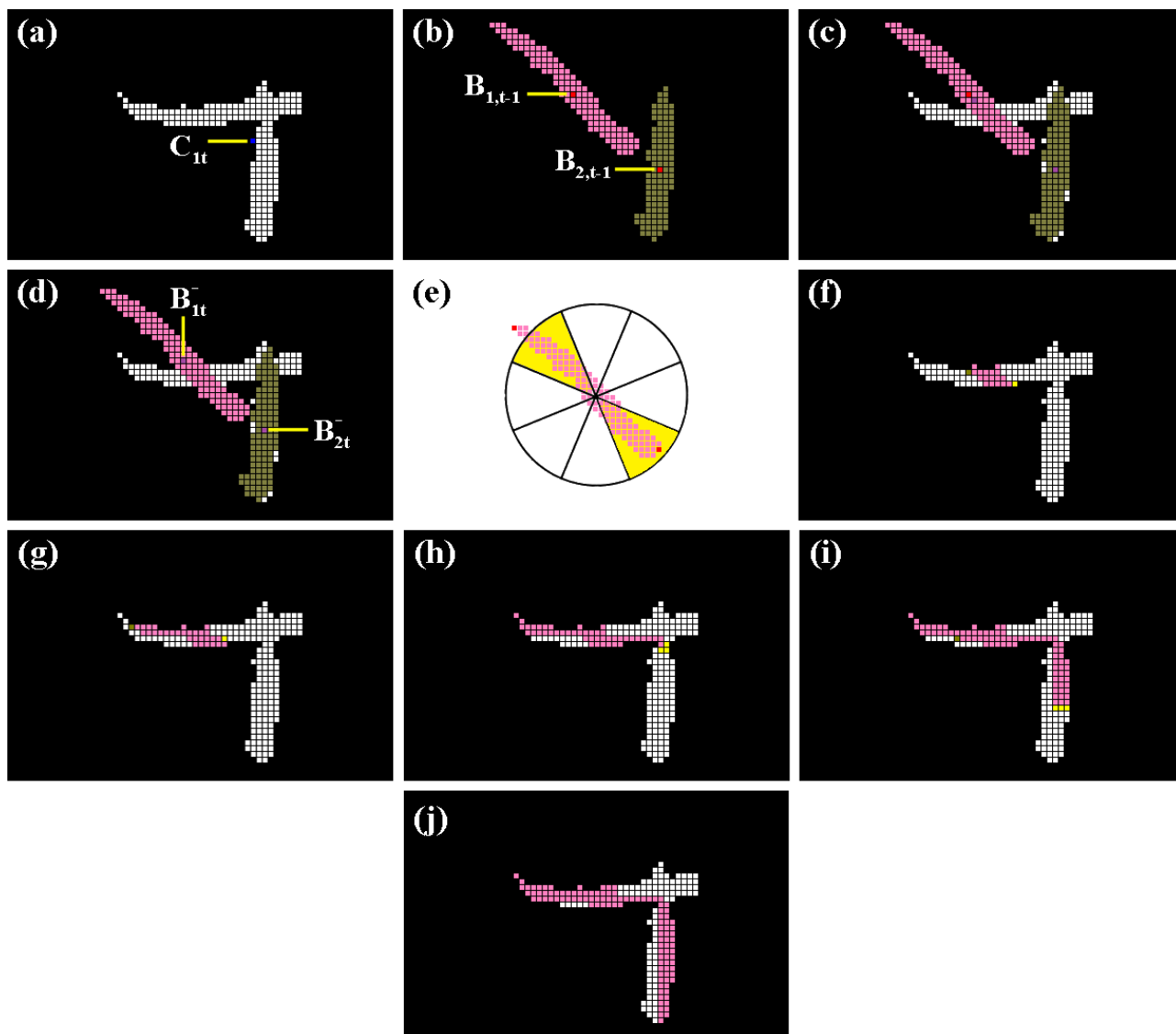


Figura 46 - Exemplo de particionamento em que um blob expande na direção errada.

Na Figura 47, são apresentados dois exemplos de bons resultados obtidos no processo de particionamento de blobs com dois peixes. Neste exemplo, após o processo de translação dos blobs $B_{1,t-1}$ e $B_{2,t-1}$ (Figura 47a-d), a região inicial definida para cada blob é pequena se comparada com a área não explorada do blob candidato C_{1t} (Figura 47e-f). Após o processo de expansão, é possível perceber que cada blob preenche a área do candidato C_{1t} correspondente ao seu próprio peixe (Figura 47g-h).

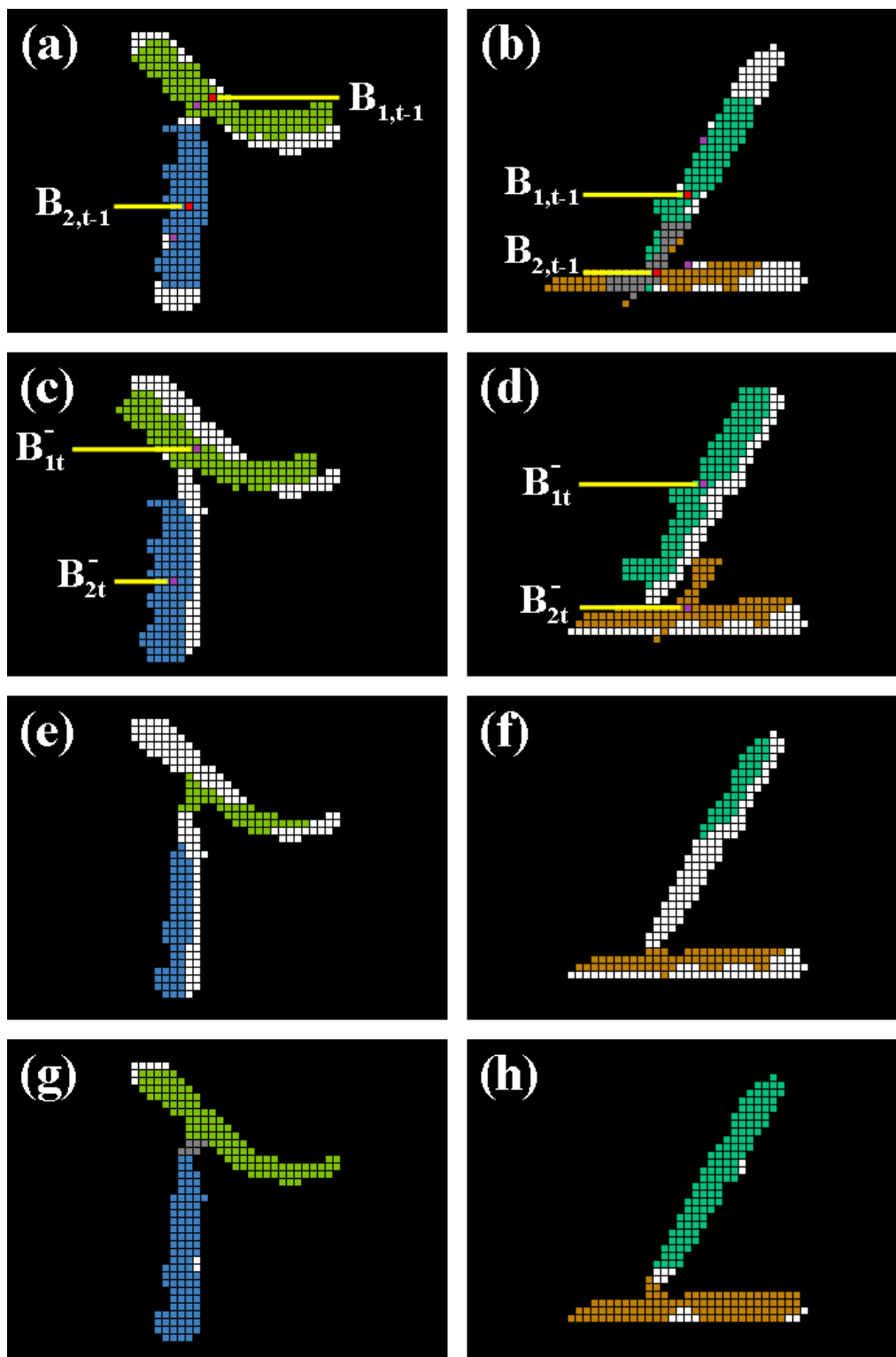


Figura 47 - Exemplo de resultados do particionamento de blobs com dois peixes.

Na Figura 48, um resultado similar é obtido em uma situação com três peixes. Neste exemplo, a Figura 48a-b ilustra o processo de translação dos blobs $B_{1,t-1}$ e $B_{2,t-1}$. As imagens (c) e (d) ilustram, respectivamente, a região inicial definida para cada blob e o resultado final do processo de expansão.

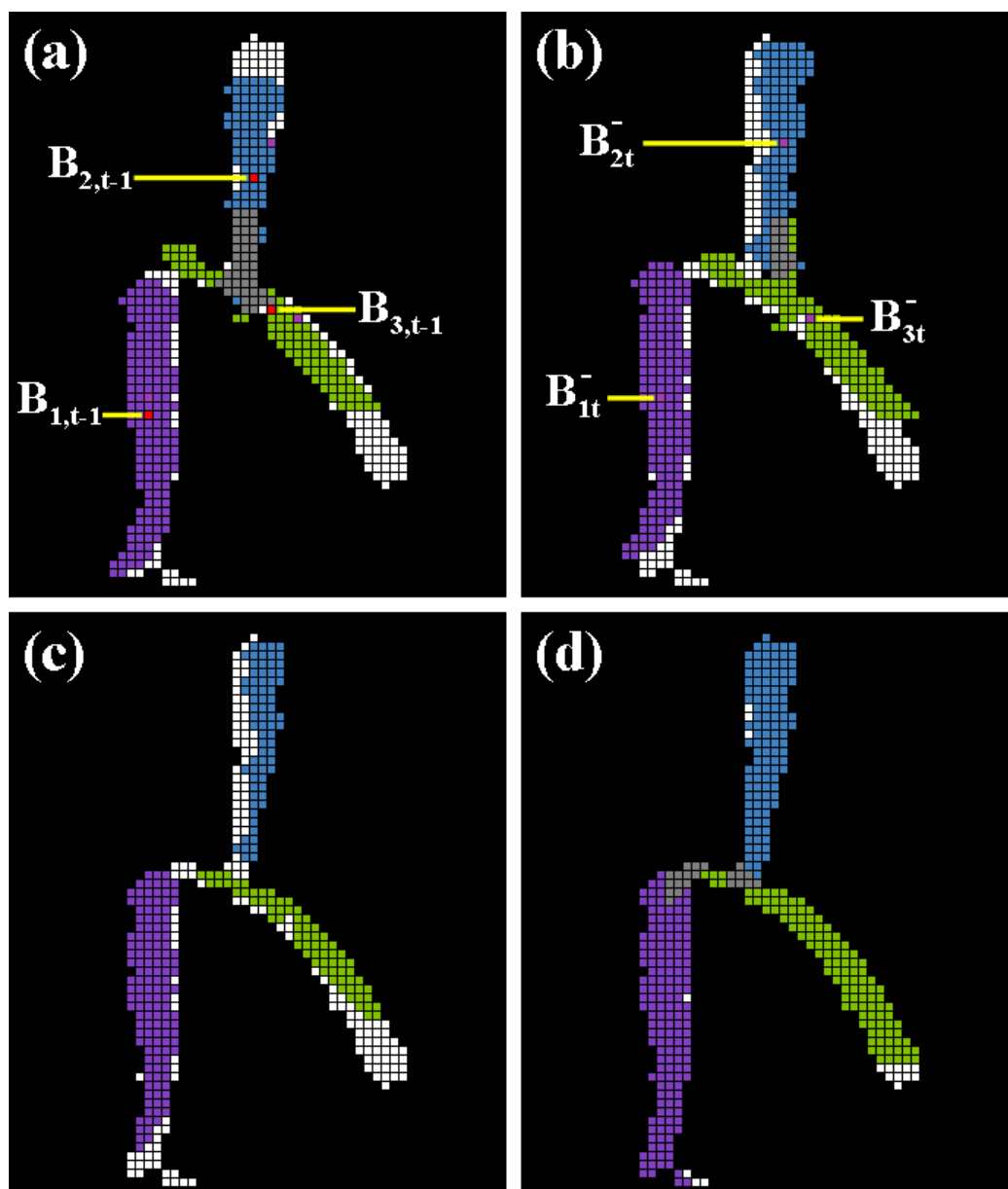


Figura 48 - Exemplo de resultados do particionamento de blobs com três peixes.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um novo algoritmo para o rastreamento de múltiplos peixes, baseado no Filtro de Kalman para manter o rastro de cada indivíduo em um aquário e no uso de operações morfológicas de dilatação para reconstruir a área de peixes envolvidos em oclusão.

Através da avaliação do método proposto, foi possível verificar que a seleção de blobs-candidatos por método de intersecção de blobs e do blob mais próximo, quando utilizados em conjunto, apresentam um aumento significativo na precisão da seleção, se comparados com estes métodos usados separadamente. Na avaliação, o método proposto foi comparado com quatro outros métodos. Três deles utilizando um único método, e um quarto método composto, em que os métodos C e INT foram combinados. O método proposto apresentou uma diminuição de 50% no número de associações incorretas, quando comparado com o segundo melhor método testado, e de aproximadamente 80%, quando comparado com o pior método.

O método de particionamento de blobs também apresentou resultados muito bons quando comparado com um *ground truth* obtido manualmente. Na avaliação, este método apresentou uma taxa de acerto de aproximadamente 98%, quando comparado com o *ground truth*. Este resultado é muito interessante, pois mostra que é possível estimar a área de um determinado peixe, com base na sua área histórica e na direção do seu movimento no *frame* anterior. Nos 2% de casos em que o algoritmo falhou, estão situações em que o peixe muda sua direção de movimento de maneira brusca.

Além das características dinâmicas dos peixes, um fator que pode prejudicar a performance do método de particionamento é a técnica de segmentação de imagens. Neste trabalho, foi utilizada uma técnica de remoção de fundo com modelo estático, em que um único modelo de fundo para o experimento inteiro. Em cenários cuja iluminação não é controlada, outra técnica de remoção de fundo seria mais adequada, como por exemplo remoção de fundo adaptativa [74].

Além disto, foi utilizado um limiar fixo para comparar o modelo de fundo com a imagem corrente. Esse limiar fixo pode prejudicar a segmentação da imagem, principalmente para detectar peixes mais jovens, que tendem a ser mais claros que peixes adultos.

No que tange à reconstrução da área referente a um peixe, em casos de oclusão, a operação de dilatação morfológica é uma técnica muito interessante pois ela permite expandir a área de um blob de forma direcionada, basicamente mudando o elemento estruturante. Até o momento não foram encontrados trabalhos que aplicassem este tipo de técnica para gerar uma nova área, apenas para diminuí-la [16].

Como trabalhos futuros, alguns pontos podem ser melhorados. Quanto à segmentação, sugere-se a avaliação de outras técnicas existentes na literatura, como, por exemplo, a técnica de remoção de fundo adaptativa já mencionada, que pode ser interessante em situação em que a iluminação pode mudar ao longo do experimento. No que se refere ao rastreamento dos peixes, outras técnicas também podem ser avaliadas, um exemplo é a técnica de Fluxo Óptico. Por fim, a identificação inicial dos peixes poderia ser realizada de forma automática através da contagem de blobs em uma sequência de imagens ou através do usuário que pode informar o sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. Farah, J. Langlois e G. Bilodeau, “RAT: Robust animal tracking,” pp. 65 - 70, 2011.
- [2] Committee on the Use of Laboratory Animals in Biomedical and Behavioral Research, Use of Laboratory Animals in Biomedical and Behavioral Research, Washington: National Academy Press, 1988, pp. 1-122.
- [3] J. Ledesma e S. McRobert, “Shoaling in juvenile guppies: The effects of body size and shoal size,” *Behavioural Processes*, vol. 77, nº 3, pp. 384-388, 2008.
- [4] O. Seppälä, A. Karvonen e E. T. Valtonen, “Shoaling behaviour of fish under parasitism and predation risk,” *Animal Behaviour*, vol. 75, nº 1, pp. 145-150, 2008.
- [5] G. Herczeg, A. Gonda e J. Merilä, “The social cost of shoaling covaries with predation risk in nine-spined stickleback, *Pungitius pungitius*, populations,” *Animal Behaviour*, vol. 77, nº 3, pp. 575-580, 2009.
- [6] G. Matthews, D. R. Davies, S. J. Westerman e R. B. Stammers, Human Performance: Cognition, Stress and Individual Differences, Psychology Press, 1 edition, 2000.
- [7] W. James, The Principles of Psychology, Dover Publications (June 1, 1950), 1890.
- [8] A. Spink, R. Tegelenbosch, M. Buma e L. Noldus, “The EthoVision video tracking system—A tool for behavioral phenotyping of transgenic mice,” *Physiology & Behavior*, vol. 73, nº 5, pp. 731-744, 2001.
- [9] N. Miller e R. Gerlai, “Quantification of shoaling behaviour in zebrafish (*Danio rerio*),” *Behavioural Brain Research*, vol. 184, nº 2, pp. 157-166, 2007.
- [10] N. Miller, K. Greene, A. Dydinski e R. Gerlai, “Effects of nicotine and alcohol on zebrafish (*Danio rerio*) shoaling,” *Behavioural Brain Research*, vol. 240, pp. 192-196, 2013.
- [11] M. Amer, E. Bilgazyev, S. Todorovic, S. Shah, I. Kakadiaris e L. Ciannelli, “Fine-grained categorization of fish motion patterns in underwater videos,” *IEEE International*

Conference on Computer Vision Workshops (ICCV Workshops), pp. 1488 - 1495, 2011.

- [12] Á. Miklósi e R. J. Andrew, "The Zebrafish as a Model for Behavioral Studies," *Zebrafish*, vol. 3, nº 2, pp. 227 - 234, 2006.
- [13] K. Faucher, E. Parmentier, C. Becco, N. Vandewalle e P. Vandewalle, "Fish lateral system is required for accurate control of shoaling behaviour," *Animal Behaviour*, vol. 79, nº 3, pp. 679-687, 2010.
- [14] L. A. Kaplan, M. Nabel, K. V. Cleef-Toedt, A. R. Proffitt e H. M. P. Jr., "Impact of benzyl butyl phthalate on shoaling behavior in *Fundulus heteroclitus* (mummichog) populations," *Marine Environmental Research*, vol. 86, pp. 70-75, 2013.
- [15] N. Reyhanian, K. Volkova, S. Hallgren, T. Bollner, P.-E. Olsson, H. Olsén e I. P. Hällström, "17 α -Ethinyl estradiol affects anxiety and shoaling behavior in adult male zebra fish (*Danio rerio*)," *Aquatic Toxicology*, vol. 105, nº 1–2, pp. 41-48, 2011.
- [16] S. Kato, T. Nakagawa, M. Ohkawa, K. Muramoto, O. Oyama, A. Watanabe, H. Nakashima, T. Nemoto e K. Sugitani, "A computer image processing system for quantification of zebrafish behavior," *Journal of Neuroscience Methods*, vol. 134, nº 1, pp. 1-7, 2004.
- [17] A. Pérez-Escudero, J. Vicente-Page, R. Hinz, S. Arganda e G. d. Polavieja, "idTracker: Tracking individuals in a group by automatic identification of unmarked animals," *Nature Methods*, vol. 11, nº 7, pp. 743 - 748, 2014.
- [18] L. Zhu e W. Weng, "Catadioptric stereo-vision system for the real-time monitoring of 3D behavior in aquatic animals," *Physiology & Behavior*, vol. 91, nº 1, pp. 106-119, 2007.
- [19] H. S. Wu, Q. Zhao, D. Zou e Y. Q. Chen, "Automated 3D trajectory measuring of large numbers of moving particles," *Opt. Express* 19, vol. 19, nº 8, pp. 7646 - 7663, 2011.
- [20] T. J. Pitcher, "Functions of Shoaling Behaviour in Teleosts," em *The Behaviour of Teleosts Fishes*, Springer, 1986, pp. 294-337.
- [21] L. Grossman, E. Utterback, A. Stewart, S. Gaikwad, K. M. Chung, C. Suciú, K. Wong, M. Elegante, S. Elkhayat, J. Tan, T. Gilder, N. Wu, J. DiLeo, J. Cachat e A. V. Kalueff,

“Characterization of behavioral and endocrine effects of LSD on zebrafish,” *Behavioural Brain Research*, vol. 214, n° 2, pp. 277-284, 2010.

- [22] J. Cote, S. Fogarty e A. Sih, “Individual sociability and choosiness between shoal types,” *Animal Behaviour*, vol. 83, n° 6, pp. 1469-1476, 2012.
- [23] L. M. Gómez-Laplaza, “Recent social environment affects colour-assortative shoaling in juvenile angelfish (*Pterophyllum scalare*),” *Behavioural Processes*, vol. 82, n° 1, pp. 39-44, 2009.
- [24] V. M. Papadakis, I. E. Papadakis, F. Lampranidou, A. Glaropoulos e M. Kentouri, “A computer-vision system and methodology for the analysis of fish behavior,” *Aquacultural Engineering*, vol. 46, pp. 53-59, 2012.
- [25] C. Buske e R. Gerlai, “Shoaling develops with age in Zebrafish (*Danio rerio*),” *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, vol. 35, n° 6, pp. 1409-1415, 2011.
- [26] N. Speedie e R. Gerlai, “Alarm substance induced behavioral responses in zebrafish (*Danio rerio*),” *Behavioural Brain Research*, vol. 188, n° 1, pp. 168-177, 2008.
- [27] J. L. Kelley, B. Phillips, G. H. Cummins e J. Shand, “Changes in the visual environment affect colour signal brightness and shoaling behaviour in a freshwater fish,” *Animal Behaviour*, vol. 83, n° 3, pp. 783-791, 2012.
- [28] C. Saverino e R. Gerlai, “The social zebrafish: Behavioral responses to conspecific, heterospecific, and computer animated fish,” *Behavioural Brain Research*, vol. 191, n° 1, pp. 77-87, 2008.
- [29] K. A. Sloman, L. Baldwin, S. McMahon e D. Snellgrove, “The effects of mixed-species assemblage on the behaviour and welfare of fish held in home aquaria,” *Applied Animal Behaviour Science*, vol. 135, n° 1-2, pp. 160-168, 2011.
- [30] A. Saxby, L. Adams, D. Snellgrove, R. W. Wilson e K. A. Sloman, “The effect of group size on the behaviour and welfare of four fish species commonly kept in home aquaria,” *Applied Animal Behaviour Science*, vol. 125, n° 3-4, pp. 195-205, 2010.
- [31] K. Volkova, N. Reyhanian, A. Kot-Wasik, H. Olsén, I. Porsch-Hällström e S. Hallgren, “Brain circuit imprints of developmental 17 α -Ethinylestradiol exposure in guppies

(*Poecilia reticulata*): Persistent effects on anxiety but not on reproductive behaviour,” *General and Comparative Endocrinology*, vol. 178, nº 2, pp. 282-290, 2012.

- [32] J. Chrásková, Y. Kaminsky e I. Krekule, “An automatic 3D tracking system with a PC and a single TV camera,” *Journal of Neuroscience Methods*, vol. 88, nº 2, pp. 195 - 200, 1999.
- [33] J. Delcourt, C. Becco, N. Vandewalle e P. Poncin, “A new automatic video multitracking system able to follow a large school,” *Proceedings of Measuring Behavior*, pp. 405-406, 2005.
- [34] J. Chen, G. Xiao, F. Gao, H. Zhou e X. Ying, “Vision-based Perceptive Framework for Fish Motion,” *International Conference on Information Engineering and Computer Science*, pp. 1 - 4, 2009.
- [35] S. Kato, K. Tamada, Y. Shimada e T. Chujo, “A quantification of goldfish behavior by an image processing system,” *Behavioural Brain Research*, vol. 80, nº 1 - 2, pp. 51-55, 1996.
- [36] J. S. Liu e R. Chen, “Sequential Monte Carlo Methods for Dynamic Systems,” *Journal of the American Statistical Association*, vol. 93, nº 443, pp. 1032 - 1044, 1998.
- [37] J. Huang, S. Kumar, M. Mitra e W.-J. Zhu, “Spatial color indexing and applications,” *International Conference on Computer Vision*, pp. 602 - 607, 1998.
- [38] B. J. Laurel, C. J. Laurel, J. A. Brown e R. S. Gregory, “A new technique to gather 3-D spatial information using a single camera,” *Journal of Fish Biology*, vol. 66, nº 2, pp. 429 - 441, 2005.
- [39] R. Y. Tsai, “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision,” em *Conference on Computer Vision and Pattern Recognition*, Miami Beach, 1986.
- [40] G. Xu e Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*, vol. 6, Springer Netherlands, 1996.
- [41] R. E. Kalman, “A New Approach to Linear Filtering and Prediction,” *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, nº Series D, pp. 35-45, 1960.

- [42] E. Maggio e A. Cavallaro, *Video Tracking - Theory and Practice*, John Wiley & Sons, 2011.
- [43] A. Jalal e V. Singh, "The state-of-the-art in visual object tracking," *International Journal of Computing and Informatics: Informatica*, vol. 36, n° 3, pp. 227-248, 2012.
- [44] F. Alsaade, "Fast and Accurate Template Matching Algorithm Based on Image Pyramid and Sum of Absolute Difference Similarity Measure," *Research Journal of Information Technology*, vol. 4, pp. 204-211, 06 2012.
- [45] L. Wang, T. Tan e W. Hu, "Face tracking using motion-guided dynamic template matching," *Asian Conference on Computer Vision*, vol. 5, 2002.
- [46] C. Veenman, M. Reinders e E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, n° 1, pp. 54-72, 2001.
- [47] D. Serby, E. Meier e L. Van Gool, "Probabilistic object tracking using multiple features," *International Conference on Pattern Recognition*, vol. 2, pp. 184 - 187, 2004.
- [48] X. Mei e H. Lin, "Robust Visual Tracking and Vehicle Classification via Sparse Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2259 - 2272, 2011.
- [49] E. Osuna, R. Freund e F. Girosi, "Training support vector machines: an application to face detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 130 - 136, 1997.
- [50] J. Xia, J. Wu, H. Zhai e Z. Cui, "Moving vehicle tracking based on double difference and camshift," em *Proceedings of the International Symposium on Information Processing*, Huangshan, 2009.
- [51] A. Jacquot, P. Sturm e O. Ruch, "Adaptive Tracking of Non-Rigid Objects Based on Color Histograms and Automatic Parameter Selection," *Seventh IEEE Workshops on Application of Computer Vision*, vol. 2, pp. 103 - 109, 2005.
- [52] X. Song e R. Nevatia, "A model-based vehicle segmentation method for tracking," *Tenth IEEE International Conference on Computer Vision*, vol. 2, pp. 1124 -1131, 2005.

- [53] A. Yilmaz, X. Li e M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, n° 11, pp. 1531 - 1536, 2004.
- [54] X. Wang, J. Sun e H.-Y. Peng, "Foreground Object Detecting Algorithm based on Mixture of Gaussian and Kalman Filter in Video Surveillance," *Journal of Computers*, vol. 8, n° 3, 2013.
- [55] R. C. Gonzalez e R. E. Woods, *Digital Image Processing*, vol. 3, Upper Saddle River, NJ, USA.: Prentice-Hall, Inc., 2006.
- [56] I. Sobel, "History and Definition of the Sobel Operator," em *ResearchGate*, 2014.
- [57] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, n° 6, pp. 679 - 698, 1986.
- [58] T. S. Ling, L. K. Meng, L. M. Kuan, Z. Kadim e A. A. B. Al-Deen, "Colour-based Object Tracking in Surveillance Application," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, p. 459, 2009.
- [59] A. Yilmaz, O. Javed e M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, n° 4, p. 13, 2006.
- [60] B. P. L. Lo e S. Velastin, "Automatic congestion detection system for underground platforms," *International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 158 - 161, 2001.
- [61] P. Arbelaez, M. Maire, C. Fowlkes e J. Malik, "From contours to regions: An empirical evaluation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2294 - 2301, 2009.
- [62] M. Piccardi, "Background subtraction techniques: a review," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099 - 3104, 2004.
- [63] G. Welch e G. Bishop, "An Introduction to the Kalman Filter," University of North Carolina, Chapel Hill, 1995.
- [64] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," *Intel Technology Journal*, 1998.

- [65] H. Bay, A. Ess, T. Tuytelaars e L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, n° 3, pp. 346 - 359, 2008.
- [66] J. A. Corrales, P. Gil, F. A. Candelas e F. Torres, "Tracking based on Hue-Saturation Features with a Miniaturized Active Vision System," In Proceedings Book of 40th International Symposium on Robotics, 2009, pp. 107 - 112.
- [67] M. Muja e D. G. Lowe, "Fast Matching of Binary Features," *Conference on Computer and Robot Vision*, pp. 404-410, 2012.
- [68] D. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, vol. 2, pp. 1150 - 1157, 1999.
- [69] T. Lindeberg, *Scale-Space Theory in Computer Vision*, vol. 256, Kluwer Academic Publishers, 1993.
- [70] R. S. Stanković e B. J. Falkowski, "The Haar wavelet transform: its status and achievements," *Computers & Electrical Engineering*, vol. 29, n° 1, pp. 25-44, 2003.
- [71] W. Jianhui, Z. Guoyun, Y. Shuai, G. Longyuan e T. Mengxia, "Study the Moving Objects Extraction and Tracking Used the Moving Blobs Method in Fisheye Image," *Pattern Recognition*, vol. 484, pp. 255 - 265, 2014.
- [72] F. Wang, X. Ren e Z. Liu, "A Robust Blob Recognition and Tracking Method in Vision-Based Multi-touch Technique," *International Symposium on Parallel and Distributed Processing with Applications*, pp. 971 - 974, 2008.
- [73] J. Serra, *Image Analysis and Mathematical Morphology*, Orlando: Academic Press, 1983.
- [74] R. Zhang e J. Ding, "Object Tracking and Detecting Based on Adaptive Background Subtraction," *Procedia Engineering*, vol. 29, pp. 1351 - 1355, 2012.