CENTRO UNIVERSITÁRIO FEI

CAIO PAVANELLI TRUFFE DE OLIVEIRA

IMPLEMENTAÇÃO E TESTES DE UMA TEORIA DE RACIOCÍNIO ESPACIAL QUALITATIVO BASEADO EM PROFUNDIDADE E MOVIMENTO

São Bernardo do Campo
2017

CAIO PAVANELLI TRUFFE DE OLIVEIRA

IMPLEMENTAÇÃO E TESTES DE UMA TEORIA DE RACIOCÍNIO ESPACIAL QUALITATIVO BASEADO EM PROFUNDIDADE E MOVIMENTO

Dissertação apresentada ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Mestre em Engenharia Elétrica. Orientado pelo Prof. Dr. Paulo Eduardo Santos.

São Bernardo do Campo
2017

Oliveira, Caio Pavanelli Truffe de .
Implementação e testes de uma teoria de raciocínio espacial qualitativo baseado em profundidade e movimento / Caio Pavanelli Truffe de Oliveira. São Bernardo do Campo, 2017.
138 p. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. Paulo Eduardo Santos.

1. Raciocínio Espacial Qualitativo. 2. Robótica Cognitiva. 3. Cálculo de Situações. 4. Perfil de Profundidade. 5. DDPC. I. Santos, Paulo Eduardo, orient. II. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

centro universitário FEI

APRESENTAÇÃO DE DISSERTAÇÃO
ATA DA BANCA EXAMINADORA

Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica

Mestrado

PGE-10

Aluno: Caio Pavanelli Truffe de Oliveira **Matrícula:** 113105-1

Título do Trabalho: Implementação e testes de uma teoria de raciocínio espacial qualitativo baseado em profundidade e movimento.

Área de Concentração: Inteligência Artificial Aplicada à Automação

Orientador: Prof. Dr. Paulo Eduardo Santos

Data da realização da defesa: 21/11/2016

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

São Bernardo do Campo, 21 / 11 / 2016.

MEMBROS DA BANCA EXAMINADORA	
Prof. Dr. Paulo Eduardo Santos	Ass.: _____
Prof. Dr. Reinaldo Augusto da Costa Bianchi	Ass.: _____
Prof. Dr. Silvio do Lago Pereira	Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO ☒ REPROVADO ☐

VERSÃO FINAL DA DISSERTAÇÃO

APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA EXAMINADORA

_____

Aprovação do Coordenador do Programa de Pós-graduação

_____

Prof. Dr. Carlos Eduardo Thomaz

Dedico este trabalho a todos aqueles que possuem a capacidade e coragem de enfrentar as dificuldades do desconhecido em busca de algo melhor.

AGRADECIMENTOS

Agradeço a meus pais, familiares, amigos e a todos que me apoiaram desde o início e acreditaram no meu potencial, me incentivando a todo instante. Agradeço também à instituição da FEI pela infraestrutura, aos professores pelo conhecimento transmitido, e em especial ao meu orientador Prof. Dr. Paulo Eduardo Santos, pela paciência e apoio mesmo após vários recomeços.

RESUMO

Levando em consideração o ponto de vista de um observador, o movimento de objetos em seu campo de visão e seu próprio movimento alteram a percepção espacial que ele possui sobre o domínio em que ele está inserido. Este trabalho investiga uma teoria de raciocínio espacial qualitativo baseada em percepções de profundidade e movimento do ponto de vista de um robô móvel. O sistema utiliza a estrutura do Cálculo de Situações para lidar com ações e suas consequências sobre movimento para processar informação espacial entre um ou mais objetos em cena. A entidade básica da teoria é um perfil de profundidade de campo gerado diretamente pelos sensores do robô, em forma simplificada através de um corte horizontal. Foi utilizada a linguagem de programação *Prolog*, e os testes foram realizados sobre sequências de ações que um robô móvel executa enquanto observa dois objetos em seu domínio. Os testes demonstraram que a teoria foi capaz de inferir todas as onze relações espaciais de transição esperadas, e que possui abertura para investigações futuras de relações espacias em outros eixos, e não apenas na horizontal.

Palavras-chave: Raciocínio Espacial Qualitativo. Robótica Cognitiva. Perfil de Profundidade. Cálculo de Situações.

ABSTRACT

Taking into consideration the point of view of an observer, the movements of objects in his field of view and his own movements alter the spatial perception he has over the domain he is part of. This work investigates a theory of qualitative spatial reasoning based on the perception of depth and motion from a robot's point of view. The system uses Situation Calculus to deal with actions and its consequences over motion to process spatial information of one or more objects in a scene. The basic entity of the theory is a depth profile generated directly from the robot's sensors, in a simplified form of a horizontal slice. The programming language used was *Prolog*, and the tests were driven over sequences of actions executed by a mobile robot while it observes two objects in its domain. The tests have shown the theory was capable of infering all the eleven expected spatial relations of transitions, and that the theory has openness for future investigation of spatial relations in other axis and not only in the horizontal axis.

Keywords: Qualitative Spatial Reasoning. Cognitive Robotics. Depth Profile. Situation Calculus.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama Conceitual de Vizinhança (CND) das relações RCC8 23
Figura 2 – Relações ROC-20 25
Figura 3 – Campo de visão do observador e perfil de profundidade da cena 32
Figura 4 – Diagrama de transições para as relações de distância, disparidade e tamanho 34
Figura 5 – Diagrama conceitual de vizinhança para transições em picos de profundidade 38
Figura 6 – Diagrama conceitual de vizinhança para transições dinâmicas para pico único 38
Figura 7 – Objetos A e B num determinado instante t 44
Figura 8 – Objetos A e B numa sequência de tempo 44
Figura 9 – Diagrama conceitual de vizinhança - CND 53
Figura 10 – Bi tangentes entre dois objetos 67
Figura 11 – Estrutura axioma *do(a,s)* 71
Figura 12 – Trajetória de testes para transições de pico individual 83
Figura 13 – Comparação de tamanho angular e disparidade de distância 88
Figura 14 – Flutuação de tamanho e distância angular dos picos de profundidade *do(a,s)* 105
Figura 15 – Perfil de profundidade com 2 picos se separandov 108
Figura 16 – Comparativo de performance entre métodos de cálculo 110
Figura 17 – Perfil de profundidade em 3 dimensões 112

SUMÁRIO

1 INTRODUÇÃO 17
2 REVISÃO BIBLIOGRÁFICA 21
2.1 RACIOCÍNIO ESPACIAL QUALITATIVO 21
2.1.1 Cálculo sobre Conexão de Regiões - RCC 22
2.1.2 Cálculo sobre Oclusão de Regiões - ROC 24
2.1.3 Relações de objetos através de transições de imagens 27
2.1.4 Raciocínio sobre profundidade e movimento 31
2.2 TRABALHOS RECENTES 43
3 REVISÃO TEÓRICA 47
3.1 CÁLCULO DE SITUAÇÕES 47
3.1.1 Raciocínio espacial qualitativo com relações topológicas 51
3.2 RACIOCÍNIO SOBRE PERFIS DINÂMICOS DE PROFUNDIDADE 53
3.2.1 Perfis de profundidade em cálculo de situações 53
3.2.2 Ações primitivas 54
3.2.3 Movimento 57
3.2.4 Atributos 58
3.2.5 Transições de pico individual 62
3.2.6 Transições entre pares de picos 64
3.2.7 Percepção de movimento 67
4 IMPLEMENTAÇÃO 69
4.1 MÉTODO COM ATUALIZAÇÃO DE SITUAÇÃO EM MEMÓRIA 69
4.1.1 Axiomas fundamentais 70
4.1.2 Axiomas auxiliares 73
4.1.3 Fluentes 74
4.2 MÉTODO SEM ATUALIZAÇÃO DE SITUAÇÃO EM MEMÓRIA 77
5 TESTES 79
5.1 TESTES DE MOVIMENTAÇÃO 80
5.2 TESTE SOBRE ATRIBUTOS 82
5.3 TESTES DE TRANSIÇÃO PARA PICO DE PROFUNDIDADE 83
5.3.1 Processo de testes de pico simples 83
5.3.2 Teste do fluente *appearing/3* 84
5.3.3 Teste do fluente *extending/3* 85
5.3.4 Teste do fluente *shrinking/3* 85
5.3.5 Teste do fluente *vanishing/3* 86
5.3.6 Teste do fluente *peak_static/3* 87
5.3.7 Resultado dos testes de transição para pico de profundidade 87
5.4 TESTES DE TRANSIÇÃO PARA PARES DE PICOS 90

5.4.1 **Processo de testes para pares de picos** 90
5.4.2 **Teste do fluente *approaching/4*** 91
5.4.3 **Teste do fluente *coalescing/4*** 94
5.4.4 **Teste do fluente *hiding/4*** 95
5.4.5 **Teste do fluente *splitting/4*** 96
5.4.6 **Teste do fluente *receding/4*** 97
5.4.7 **Teste do fluente *two_peak_static/4*** 98
5.4.8 **Resultado dos testes de transição para pares de picos** 98
6 **RESULTADOS E DISCUSSÃO** 107
7 **CONCLUSÃO** 111
REFERÊNCIAS 114
APÊNDICE A – Axiomas DDPC 117
APÊNDICE B – Tabela de testes sentido anti-horário 131

1 INTRODUÇÃO

Este trabalho foi motivado pelos avanços recentes na área de veículos inteligentes. Grandes empresas do setor automotivo têm investido de forma privada em soluções que abrangem desde o aumento da segurança de seus veículos através de ferramentas preventivas e corretivas até o controle total da direção por pilotagem automática. Tais soluções utilizam conjuntos de ferramentas de diversas áreas do conhecimento como visão computacional, robótica cognitiva e inteligência computacional.

O estudo aqui realizado está contido dentro da área de Robótica Cognitiva que, conforme Levesque e Lakemeyer (2008), tem o objetivo de equipar um robô com uma inteligência de alto nível capaz de gerar ações baseadas num grupo de informações que compreendem o seu conhecimento e objetivo. Diferentes vertentes impulsionam esta área do conhecimento como planejamento (TRAVERSO; GHALLAB; NAU, 2004 **apud** LEVESQUE; LAKEMEYER, 2008) e aprendizado (MICHALSKI; CARBONELL; MITCHELL, 2013).

Além disso, este trabalho também faz parte da área de Raciocínio Espacial Qualitativo, que tem como desafio promover o cálculo, representação e raciocínio sobre entidades espaciais sem recorrer aos recursos tradicionais da computação quantitativa (COHN; RENZ, 2008). São parte desta área estudos sobre sistemas de informação geográfica como em (EGENHOFER; MARK, 1995 **apud** LEVESQUE; LAKEMEYER, 2008) e Chevriaux, Saux e Claramunt (2005), navegação de robôs (KUIPERS; BYUN, 1991; WAGNER; HÜBNER, 2004), análise qualitativa de cenas em vídeo (FERNYHOUGH; COHN; HOGG, 2000 **apud** LEVESQUE; LAKEMEYER, 2008), linguagem natural (ZEE; SLACK, 2003 **apud** LEVESQUE; LAKEMEYER, 2008) entre outros. No Capítulo 2 foi realizada uma revisão bibliográfica mais detalhada dos principais trabalhos relacionados à esta pesquisa.

O objetivo deste trabalho é implementar o formalismo definido em (SOUTCHANSKI; SANTOS, 2008). Esta teoria foi escolhida por ter sido idealizada com o objetivo de que ela evoluísse para um sistema de raciocínio automatizado capaz de inferir planos de veículos que estariam ao redor de um carro inteligente, baseando-se no conhecimento próprio de suas ações e nas observações realizadas através de seus sensores.

Assim como em Santos (2007), considera-se que um robô móvel possua um sistema de visão computacional formado por uma câmera estéreo ou laser, que seja capaz de gerar uma descrição do seu ponto vista por meio da informação de disparidade de distância entre o robô e todos os pontos visíveis a sua frente (limitado pela resolução do equipamento). Este conjunto de dados é chamado de *perfil de profundidade*. Utiliza-se também a premissa de que todos os objetos do mundo são cilindros. Por este motivo, considera-se apenas a faixa central do perfil de profundidade, à meia altura do robô, formando um conjunto de dados em duas dimensões que seria suficiente para descrever objetos ao seu redor. O processamento destes dados permite a extração de informações como tamanho angular, distância angular até a borda esquerda do

campo de visão e disparidade de profundidade para o ponto de vista do observador, que serão tratados como atributos de cada pico de profundidade.

A teoria de Soutchanski e Santos (2008) também propõe um raciocínio lógico sobre as ações que acontecem no mundo. Para tanto, é utilizado o *Cálculo de Situações* (REITER, 2001). No Cálculo de Situações é utilizado o conceito de que cada instante de tempo é justificado por *situações*, que são sequências finitas de ações. Diferentemente do conceito de estados, situações são comparáveis apenas pelo histórico completo de ações até a *situação inicial*, e não apenas pelos valores das variáveis que descrevem o domínio num determinado instante de tempo. Entende-se por situação inicial um conjunto mínimo de informações que permitem a inicialização e o funcionamento do sistema, e que todas as ações são subsequentes à situação inicial.

O conjunto de informações obtidas pelo sistema de sensoriamento do robô e o raciocínio sobre ações do mundo permitem que ele gere hipóteses sobre relações espaciais dos objetos presentes em seu campo de visão. Cada objeto é representado por um pico no perfil de profundidade. A percepção de tamanho e localização destes picos se alteram conforme ações são realizadas no mundo. Pela observação de tais mudanças, são realizados cálculos que determinam se um pico de profundidade está crescendo (*extending*), diminuindo (*shrinking*), aparecendo (*appearing*), desaparecendo (*vanishing*) ou se está estático (*peak_static*).

Quando existir mais de um objeto no campo de visão do robô móvel, e consequentemente mais de um pico no perfil de profundidade, os dados dos atributos de cada pico permitem que sejam geradas hipóteses sobre relações espaciais entre cada par de picos. Dois picos de profundidade podem se aproximar (*approaching*), se afastar (*receding*), se fundir (*coalescing*), se separar (*splitting*), estar estáticos (*two_peak_static*) ou um objeto pode ocluir o outro (*hiding*), o que significa que haveria apenas um pico, porém devido ao histórico de ações sabe-se que tratam-se na verdade de dois. No Capítulo 3 a teoria de Soutchanski e Santos (2008) está descrita em detalhes contendo desde as relações mencionadas até as funções auxiliares.

Foi desenvolvido um código em Prolog que espelha as definições da teoria de (SOUTCHANSKI; SANTOS, 2008). Os detalhes referentes à implementação estão descritos no Capítulo 4, assim como os testes realizados. Foram geradas sequências de ações que colocassem um robô móvel em pontos de vista que fossem capazes de observar todas as relações espaciais previstas. As ações foram separadas em 3 blocos. No primeiro bloco a intenção foi observar exclusivamente a mudança de percepção do robô em relação aos atributos de um pico de profundidade. No segundo bloco a intenção foi observar as transições para pico individual. No terceiro bloco a intenção foi observar as transições entre um par de picos. O código desenvolvido foi publicado em *https://github.com/pava197/qsrTdm-FEI*.

Nos Capítulos 5 e 6 foram descritos os resultados obtidos e as considerações finais. Em síntese, a teoria de Soutchanski e Santos (2008) atendeu às expectativas quanto ao raciocínio sobre ações e mudanças que ocorrem no mundo e todas as relações espaciais foram observadas. É importante notar que os testes foram realizados em ambiente controlado e através de simu-

lação. Existe também a possibilidade de continuidade da pesquisa como a investigação de um sistema de visão computacional que trate dados reais de um sensor para alimentação do sistema e extensão da teoria para três dimensões.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo tem como objetivo contextualizar o presente trabalho na literatura atual descrevendo as teorias e conceitos nos quais ele se baseia. A teoria principal relacionada é a teoria chamada *Theory of Depth and Motion* ($\mathcal{TDM}$) definida em Soutchanski e Santos (2008) de raciocínio espacial qualitativo sobre a percepção de profundidade e movimento de objetos físicos. O trabalho descrito pertence à área de robótica cognitiva, e tem como objetivo munir um robô com habilidades cognitivas de alto nível através da representação lógica de objetos do mundo e inferência de relações espaciais que descrevem a atual situação do domínio. Na sequência serão citados os trabalhos fundamentais de maior relevância para esta pesquisa.

2.1 RACIOCÍNIO ESPACIAL QUALITATIVO

Raciocínio Espacial Qualitativo (REQ) é a área da inteligência artificial que tem como foco o estudo e a representação lógica espacial de entidades de um domínio e suas relações através de uma linguagem de alto nível. Em Cohn e Renz (2008) foi desenvolvido um estudo detalhado sobre as diferentes vertentes da área de representação e raciocínio espacial qualitativo. Ontologicamente as entidades primitivas do domínio podem ser pontos, segmentos de linha (MORATZ; RENZ; WOLTER, 2000), regiões (RANDELL; CUI; COHN, 1992), entre outros (CHEN et al, 2015). Uma das premissas desta área de estudo é a definição de relações entre as entidades do domínio. Tais relações são comumente utilizadas se forem mutuamente exclusivas e disjuntas par a par. Isto significa que dado um conjunto de relações possíveis no mundo, a relação entre duas entidades será necessariamente apenas uma. Cada conjunto de relações que atenda este critério é uma teoria diferente como por exemplo o Cálculo sobre Conexão de Regiões (RCC) (RANDELL; CUI; COHN, 1992). As relações em sua maioria são binárias, porém quando relações de direção e orientação são tratadas, há a necessidade de informação adicional como um objeto primário, um objeto de referência ou um campo de referência onde os objetos de encontram (COHN; RENZ, 2008). Distância e tamanho podem tanto ser representados em uma escala absoluta ou relativa.

Diversas pesquisas impulsionam a área de raciocínio espacial qualitativo. Outro exemplo que pode ser citado são os sistemas de informação geográfica (GIS), pertencentes à subárea de Geografia Ingênua (*Naive Geography*) (EGENHOFER; MARK, 1995). Nesta subárea, formalismos espaciais são definidos para facilitar a interação com o usuário do sistema e para gerar inferências que responderiam em uma linguagem natural comparáveis à forma que uma pessoa o faria, o que aumenta a confiabilidade do usuário para utilização de sistemas de informação geográfica, e dispensa extensivos treinamentos que ainda são necessários.

Dentro da subárea de navegação de robôs, foi desenvolvido por Kuipers e Byun (1991) um sistema qualitativo para exploração e geração de um mapa referente ao ambiente em que o

robô se encontra. Já Chen e Birchfield (2006), definiu um formalismo qualitativo para que um robô possa repetir uma trajetória cuja qual ele tenha sido treinado.

Além disso, também foi explorada qualitativamente a análise de sequências de cenas em vídeo. Em (FERNYHOUGH; COHN; HOGG, 2000), foi proposto um sistema capaz gerar modelos de eventos através do acompanhamento do contorno de objetos em cada quadro. Criou-se uma região relevante no vídeo por onde os objetos se movem e por um método estatístico criou-se modelos qualitativos de eventos que permitiram acompanhar eventos específicos ou encontrar eventos incomuns. Em Nath e Hazarika (2015) foi proposta uma combinação de raciocínio sobre diagramas com representação espacial e temporal qualitativa para detecção de eventos de movimento em vídeo.

Em Kreutzmann (2013) foi proposta uma linguagem baseada em lógica espacial e temporal qualitativa para representar regras de navegação. O objetivo foi criar um sistema capaz de auxiliar operadores a seguir e a cumprir regulamentações sobre prevenção de colisão entre embarcações.

2.1.1 Cálculo sobre Conexão de Regiões - RCC

O Cálculo sobre Conexão de Regiões é um conjunto de axiomas de lógica de primeira ordem sobre relações espaciais baseadas na relação primitiva de conectividade entre duas regiões (C/2). A conectividade entre as regiões é verdadeira caso exista pelo menos um ponto em comum entre uma região x e uma região y. Os axiomas que formam a base desta teoria são:

a) DC(x,y): "x está desconectado de y";

b) EC(x,y): "x está externamente conectado à y";

c) PO(x,y): "x sobrepõe parcialmente y";

d) TPP(x,y): "x é parte própria tangencial de y";

e) EQ(x,y): "x é idêntico à y";

f) NTPP(x,y): "x é parte própria não tangencial de y";

g) P(x,y): "interpreta-se que x é parte de y";

h) PP(x,y): "x é parte própria de y";

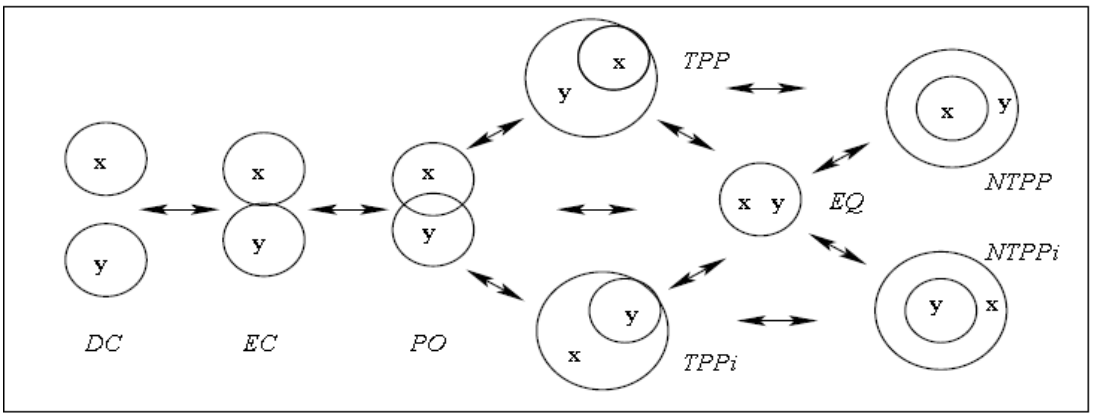
i) O(x,y): "x sobrepõe y";

j) DR(x,y): "x é discreto de y";

k) TPPi(x,y): "y é parte própria tangencial de x";

l) NTPPi(x,y): "y é parte própria não tangencial de x".

Figura 1 – Diagrama Conceitual de Vizinhança (CND) das relações RCC8



Fonte: Randell; Cui; Cohn, 1992

As relações DC, EC, PO, TPP, EQ, NTPP, TPPi, NTPPi formam um conjunto exaustivo e disjunto par a par chamado de RCC8.

Na Figura 1 pode-se observar a continuidade nas transições de estado no diagrama conceitual de vizinhança (CND). Entre dois vértices do gráfico existe apenas uma única relação possível entre os objetos, ou seja, cada vez que o mundo sofre a influência de uma ação que resulta numa transição de estados, a relação do estado seguinte é sempre uma relação vizinha ou próxima.

As definições das relações apresentadas resultam em interpretações sobre os objetos observados que podem ser:

$$DC(x,y) \equiv_{def.} \neg C(x,y) \tag{1}$$

Interpretada como x está desconectado de y.

$$P(x,y) \equiv_{def.} \forall z[C(z,x) \rightarrow C(z,y)] \tag{2}$$

Interpretada como x é parte de y.

$$PP(x,y) \equiv_{def.} P(x,y) \wedge \neg P(y,x) \tag{3}$$

Interpretada como x é parte própria de y.

$$EQ(x,y) \equiv_{def.} P(x,y) \wedge P(y,x) \tag{4}$$

Interpretada como x é idêntico a y.

$$O(x,y) \equiv_{def.} \exists z[P(z,x) \wedge P(z,y)] \tag{5}$$

Interpretada como x sobrepõe y.

$$PO(x,y) \equiv_{def.} O(x,y) \wedge \neg P(x,y) \wedge \neg P(y,x) \tag{6}$$

Interpretada como x sobrepõe parcialmente y.

$$DR(x,y) \equiv_{def.} \neg O(x,y) \tag{7}$$

Interpretada como x é discreto a y.

$$EC(x,y) \equiv_{def.} C(x,y) \wedge \neg O(x,y) \tag{8}$$

Interpretada como x está externamente conectado a y.

$$TPP(x,y) \equiv_{def.} PP(x,y) \wedge \exists z[EC(z,x) \wedge EC(z,y)] \tag{9}$$

Interpretada como x é parte própria tangencial de y.

$$NTPP(x,y) \equiv_{def.} PP(x,y) \wedge \neg\exists z[EC(z,x) \wedge EC(z,y)] \tag{10}$$

Interpretada como x é parte própria não tangencial de y.

$$Pi(x,y) \equiv_{def.} P(y,x) \tag{11}$$

Interpretada como y é parte de x.

$$PPi(x,y) \equiv_{def.} PP(y,x) \tag{12}$$

Interpretada como y é parte própria de x.

$$TPPi(x,y) \equiv_{def.} TPP(y,x) \tag{13}$$

Interpretada como y é parte própria tangencial de x.

$$NTPPi(x,y) \equiv_{def.} NTPP(y,x) \tag{14}$$

Interpretada como y é parte própria não tangencial de x.

O RCC8 é a base fundamental para o presente trabalho.

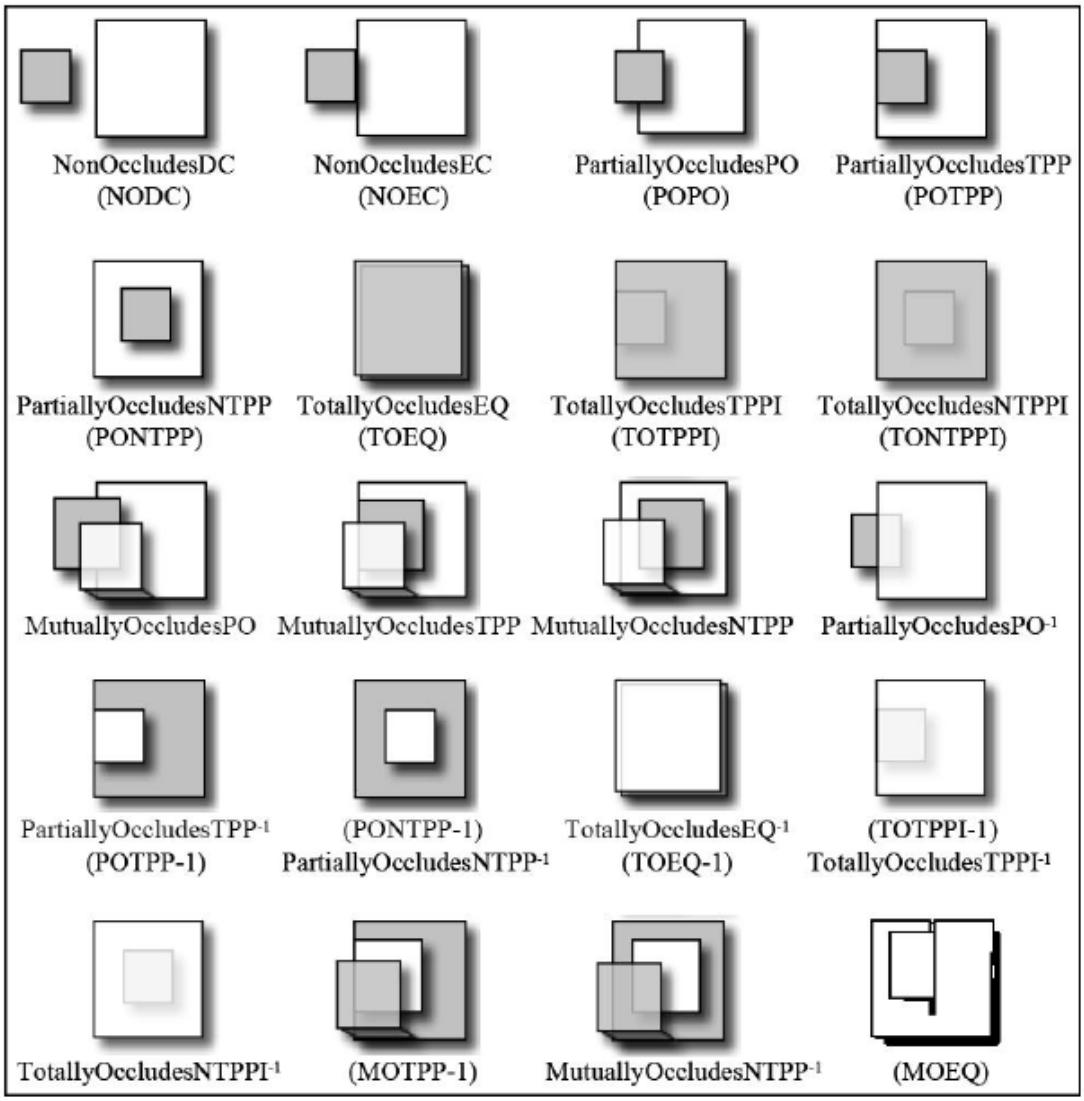
2.1.2 Cálculo sobre Oclusão de Regiões - ROC

O Cálculo sobre Oclusão de Regiões (ROC) Randell, Witkowski e Shanahan (2001) estende o RCC8 de modo que esta teoria permite estudar as relações de oclusão espacial e paralaxe entre entidades do domínio a partir de um ponto de vista. Tais relações tornam possível inferir o porquê de um objeto não estar visível e onde está o contorno deste objeto.

A oclusão espacial ocorre pela interposição de objetos a partir de um ponto de vista, seja por movimentação dos objetos ou do observador. O ROC também conhecido por ROC-20 Randell e Witkowski (2002) define 20 alinhamentos de objetos, e também obedece à continuidade de transições presente no RCC8, ou seja, cada vez que uma ação do mundo gera uma

transição de estado, este novo estado será vizinho ou próximo. Na Figura 2 podemos observar uma ilustração de todas as relações do ROC-20.

Figura 2 – Relações ROC-20



Fonte: Randell; Witkowski, 2002

O ROC define uma distinção entre a região ocupada por um corpo no mundo e sua projeção em relação a um observador. Formalmente temos a função $region(x)$, que representa a região ocupada por x, e a função $image(x,v)$, que representa a imagem de x em relação ao ponto de vista v (RANDELL; WITKOWSKI, 2002).

A principal relação que define a oclusão no ROC é a de oclusão total, que é irreflexiva e transitiva, $TotallyOccludes(x,y,v)$, onde x oclui totalmente y do ponto de vista v (RANDELL; WITKOWSKI; SHANAHAN, 2001). A justificativa teórica para esta relação é a de que não existe um ponto em y que se possa traçar uma reta até o ponto de vista do observador, que não atravesse um ponto em x. As demais relações da teoria são graduações e variações possíveis de acordo com o posicionamento e forma dos objetos observados. Seguem abaixo alguns axiomas que constituem a teoria:

$$\forall x \forall v[\neg TotallyOccludes(x,x,v)], \tag{15}$$

isto é, para todo objeto x, x não pode ocluir a si mesmo.

$$\forall x \forall y \forall z \forall v[[TotallyOccludes(x,y,v) \wedge TotallyOccludes(y,z,v)] \rightarrow TotallyOccludes(x,z,v)], \tag{16}$$

isto é, se x oclui totalmente y, e y oclui totalmente z, então x oclui totalmente z.

$$\forall x \forall y \forall z \forall v[[TotallyOccludes(x,y,v) \wedge P(region(z),region(y))] \rightarrow TotallyOccludes(x,z,v)], \tag{17}$$

isto é, se x oclui totalmente y, então x oclui totalmente qualquer parte de y.

$$\forall x \forall y \forall v[TotallyOccludes(x,y,v) \rightarrow \forall z[P(region(z),region(y))] \rightarrow \neg TotallyOccludes(z,x,v)], \tag{18}$$

isto é, se x oclui totalmente y, então nenhuma parte de y oclui totalmente x.

$$\forall x \forall y \forall v[TotallyOccludes(x,y,v) \rightarrow \forall z \forall u[[P(region(z),region(x)) \wedge P(region(u),region(y))] \rightarrow \neg TotallyOccludes(u,z,v)]], \tag{19}$$

isto é, se x oclui totalmente y, nenhuma parte de y oclui totalmente alguma parte de x.

A conjunto completo de relações ROC pode ser visto em (RANDELL; WITKOWSKI; SHANAHAN, 2001). Outras relações de oclusão que podem ser definidas além da oclusão total são oclusão parcial, mútua e não oclusão, através das funções $PartiallyOccludes(x,y,v)$, onde x oclui parcialmente y do ponto de vista v, $MutuallyOccludes(x,y,v)$, onde x e y se ocluem mutuamente do ponto de vista v, e $NonOccludes(x,y,v)$, onde x não oclui y do ponto de vista v.

As relações de oclusão total, parcial, mútua e situação de não oclusão do ROC estão relacionadas diretamente com as relações do RCC e formam o conjunto de axiomas ROC-20 ilustrados na Figura 2. A formalização das relações obedece às Fórmulas 20 e 21.

$$\Phi\Psi(x,y,v) \equiv_{def.} \Phi(x,y,v) \wedge \Psi(image(x,v),image(y,v)). \tag{20}$$

$$X\Psi^{-1}(x,y,v) \equiv_{def.} X(y,x,v) \wedge \Psi(image(y,v),image(x,v)). \tag{21}$$

Onde,

se Φ = $NonOccludes$, então $\Psi \in \{DC,EC\}$,

se Φ = $TotallyOccludes$, então $\Psi \in \{EQ,TPPi,NTPPi\}$,
se Φ = $PartiallyOccludes$, então $\Psi \in \{PO,TPP,NTPP\}$,
se Φ = $MutuallyOccludes$, então $\Psi \in \{PO,EQ,TPP,NTPP\}$,
e onde,
se X = $TotallyOccludes$, então $\Psi \in \{EQ,TPPi,NTPPi\}$,
se X = $PartiallyOccludes$, então $\Psi \in \{PO,TPP,NTPP\}$,
se X = $MutuallyOccludes$, então $\Psi \in \{TPP,NTPP\}$.

As relações definidas pelo ROC-20 são base para a teoria estudada neste trabalho. A forma como são inferidas relações de oclusão através de um conjunto de outras relações espaciais se assemelha à forma em que é definida a relação $hiding$, que será detalhada no Capítulo 3.

2.1.3 Relações de objetos através de transições de imagens

Em Santos e Shanahan (2002) foi proposto um modelo baseado nas teorias RCC e ROC que propõe uma caracterização dinâmica de oclusão, baseada na informação direta obtida do sistema de visão de um robô móvel.

O processo de interpretação da informação obtida do sistema de sensoriamento de um robô foi divido em etapas (SANTOS; SHANAHAN, 2002). Em primeiro lugar, cenas do mundo são representadas como regiões espaciais em duas dimensões. As relações entre tais regiões são caracterizadas por uma linguagem lógica similar à presente no RCC. A segunda etapa leva em consideração uma sequência cronológica de cenas do mundo que formam uma representação de suas transições em termos de predicados dinâmicos sobre imagens de cada objeto. E, por fim, a última etapa consiste em gerar hipóteses sobre relações espaciais dinâmicas que justificam as transições percebidas na informação sensorial obtida pelo robô.

As relações espaciais em Santos e Shanahan (2002) são dependentes de uma função primitiva de distância. Assume-se que essa informação seja obtida diretamente dos sensores do robô através de técnicas de visão computacional, não inerentes à teoria, que compreende a menor distância entre dois pontos pertencentes aos limites de duas regiões. No entanto, de posse dessa informação, através do grau de distanciamento de duas regiões, é possível inferir as relações $DC(x,y)$, onde x está desconectado de y, $EC(x,y)$, onde x está externamente conectado a y, e $Co(x,y)$, onde x está fundido a y. Os axiomas referentes às funções mencionadas estão representados pelas Fórmulas (22), (23) e (24).

$$DC(x,y) \leftrightarrow (dist(x,y) > \delta). \tag{22}$$

$$EC(x,y) \leftrightarrow (dist(x,y) \leq \delta) \wedge (dist(x,y) \neq 0). \tag{23}$$

$$Co(x,y) \leftrightarrow (dist(x,y) = 0). \tag{24}$$

Onde δ é determinado de acordo com o domínio. No caso de um robô móvel, pode ser considerado como a largura do robô.

A representação literal das imagens dos objetos visualizados pelo robô é realizada pela função $i/3$ (RANDELL; WITKOWSKI; SHANAHAN, 2001). A função $i(x,v,t)$ representa a imagem de x do ponto de vista v no instante de tempo t, ou seja, $i/3$ representa um mapeamento dos objetos visíveis, pontos de vista e instantes no tempo para regiões espaciais nas imagens dos sensores do robô móvel. Ao assumir que os resultados gerados por esta função sejam as relações Co, DC e EC, é possível através desta informação definir os predicados dinâmicos: $approaching(i(a,v,t),i(b,v,t))$, onde as imagens de a e b estão se aproximando de acordo com o ponto de vista v no instante de tempo t; $receding(i(a,v,t),i(b,v,t))$, onde as imagens de a e b estão se afastando de acordo com o ponto de vista v no instante de tempo t; $coalescing(i(a,v,t),i(b,v,t))$, onde as imagens de a e b estão se unificando de acordo com o ponto de vista v no instante de tempo t; $splitting(i(a,v,t),i(b,v,t))$, onde as imagens de a e b estão se separando de acordo com o ponto de vista v no instante de tempo t. As Fórmulas (25) a (28) que formalizam os axiomas destas funções, fazem uso do predicado $between/3$ Gärdenfors (2004) para manter uma relação de ordem entre pontos de vista.

A relação $between(x, y, z)$ é interpretada como x está entre y e z. É aceitável considerar que x, y e z sejam colineares porque os instantes de tempo t_1 e t_2 correspondem a um curto intervalo tempo e por este motivo não há perda de generalização ao assumir colinearidade entre pares de pontos de vista observados em t_1 e t_2. Vale notar que as Fórmulas (25) a (28) não compreendem bi-implicação porque tratam-se de hipóteses, diferentemente das relações $DC(x,y)$, $EC(x,y)$ e $Co(x,y)$. Por este motivo a definição seguinte não pode simplesmente ser definida exclusivamente pela função $dist/2$.

$$\begin{aligned} approaching(i(a,v,t),i(b,v,t)) \rightarrow \exists t_1 \exists t_2 \exists v_1 \exists v_2 (t_1 < t) \wedge (t < t_2) \wedge \\ between(v,v_1,v_2) \wedge DC(i(a,v_1,t_1),i(b,v_1,t_1)) \wedge \neg Co(i(a,v_2,t_2),i(b,v_2,t_2)) \wedge \\ (dist(i(a,v_1,t_1),i(b,v_1,t_1)) > dist(i(a,v_2,t_2),i(b,v_2,t_2))). \end{aligned} \tag{25}$$

A Fórmula (25) expressa que se duas imagens estão se aproximando no instante de tempo t, então elas estão desconectadas num instante de tempo t_1 anterior a t, não estão fundidas num instante t_2 posterior a t, e a distância entre as imagens é maior em t_1 do que em t_2. A condição de que as imagens não estão fundidas no instante de tempo seguinte garante que o Axioma $approaching/2$ não inclua a possibilidade de $coalescing/2$.

$$\begin{aligned} coalescing(i(a,v,t),i(b,v,t)) \rightarrow \exists t_1 \exists t_2 \exists v_1 \exists v_2 (t_1 < t) \wedge (t < t_2) \wedge \\ between(v,v_1,v_2) \wedge (EC(i(a,v_1,t_1),i(b,v_1,t_1)) \vee DC(i(a,v_1,t_1),i(b,v_1,t_1))) \wedge \\ Co(i(a,v_2,t_2),i(b,v_2,t_2)). \end{aligned} \tag{26}$$

Se duas imagens estão fundidas no instante de tempo t, então elas estão externamente conectadas (EC) ou desconectadas (DC) num instante t_1 anterior a t e estão fundidas (Co) no instante t_2 posterior a t. As interpretações das Fórmulas (27), (28) e (29) são análogas.

$$\begin{aligned} splitting(i(a,v,t),i(b,v,t)) \rightarrow \exists t_1 \exists t_2 \exists v_1 \exists v_2 (t_1 < t) \wedge (t < t_2) \wedge \\ between(v,v_1,v_2) \wedge Co(i(a,v_1,t_1),i(b,v_1,t_1)) \wedge (EC(i(a,v_2,t_2),i(b,v_2,t_2)) \vee \\ DC(i(a,v_2,t_2),i(b,v_2,t_2))). \end{aligned} \tag{27}$$

$$\begin{aligned} receding(i(a,v,t),i(b,v,t)) \rightarrow \exists t_1 \exists t_2 \exists v_1 \exists v_2 (t_1 < t) \wedge (t < t_2) \wedge \\ between(v,v_1,v_2) \wedge (EC(i(a,v_1,t_1),i(b,v_1,t_1)) \vee DC(i(a,v_1,t_1),i(b,v_1,t_1))) \wedge \\ (dist(i(a,v_1,t_1),i(b,v_1,t_1)) < dist(i(a,v_2,t_2),i(b,v_2,t_2))). \end{aligned} \tag{28}$$

$$\begin{aligned} static(i(a,v,t),i(b,v,t)) \rightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge \\ (dist(i(a,v,t_1),i(b,v,t_1)) = dist(i(a,v,t_2),i(b,v,t_2))). \end{aligned} \tag{29}$$

Através das relações entre imagens definidas acima, também foram propostas relações dinâmicas sobre os objetos observados: $getting_closer(a,b,v,t)$, onde os objetos a e b estão se aproximando no instante de tempo t do ponto de vista v; $ap_getting_closer(a,b,v,t)$, onde os objetos a e b estão aparentemente se aproximando no instante de tempo t do ponto de vista v devido movimento do observador; $getting_further(a,b,v,t)$, onde os objetos a e b estão se afastando no instante de tempo t do ponto de vista v; $ap_getting_further(a,b,v,t)$, onde os objetos a e b estão aparentemente se afastando no instante de tempo t do ponto de vista v devido movimento do observador; $occluding(a,b,v,t)$, onde um dos objetos a e b está se movendo para frente do outro no instante de tempo t do ponto de vista v; $touching(a,b,v,t)$, onde os objetos a e b estão se tocando no instante de tempo t do ponto de vista v; $static(a,b,v,t)$, onde os objetos a e b estão estáticos do ponto de vista v. Estas relações são hipóteses de possíveis relações entre objetos que explicam as relações observadas entre regiões de imagens, por este motivo não foi descrito uma definição formal para cada um.

Em seguida, com adição do predicado $located(a,v,t)$, que representa o fato do corpo físico a estar localizado em v no instante de tempo t, e que o robô móvel possui a capacidade de se localizar no ambiente em que ele se encontra, a relação entre os predicados sobre imagens e sobre objetos físicos é formalizada pelos Axiomas (30) a (33). Neste grupo de axiomas a constante $ROBOT$ representa o robô móvel. É interessante notar que os Axiomas (25) a (26) representam relações espaciais que explicam observações de imagens, e os Axiomas (30) a (33) representam hipóteses para tais relações.

$$approaching(i(a,v,t),i(b,v,t)) \leftarrow located(ROBOT,v,t) \wedge$$

$$(getting_closer(a,b,v,t) \vee ap_getting_closer(a,b,v,t)). \tag{30}$$

$$\begin{aligned} coalescing(i(a,v,t),i(b,v,t)) \leftarrow located(ROBOT,v,t) \wedge \\ (occluding(a,b,v,t) \vee touching(a,b,v,t)). \end{aligned} \tag{31}$$

$$\begin{aligned} (splitting(i(a,v,t),i(b,v,t)) \vee receding(i(a,v,t),i(b,v,t))) \leftarrow located(ROBOT,v,t) \wedge \\ (getting_further(a,b,v,t) \vee ap_getting_further(a,b,v,t)). \end{aligned} \tag{32}$$

$$static(i(a,v,t),i(b,v,t)) \leftarrow located(ROBOT,v,t) \wedge static(a,b,v,t). \tag{33}$$

Por fim, foram introduzidos conceitos de explicação de dados sensoriais através de abdução utilizando as Fórmulas (25) à (33). E predição de configuração futura através de dedução.

Abdução é definida como o processo de explicação de uma sentença Γ através de uma fórmula Δ não vazia, de forma que dada uma teoria de base Σ , Γ é uma consequência lógica de $\Sigma \cup \Delta$ (SHANAHAN, 1996), desde que seja consistente. Vale notar que explicações aceitáveis compreendem apenas um conjunto de predicados considerados abdutíveis. Levando-se em consideração o arcabouço definido nesta seção, o conjunto de Fórmulas (22), (23) e (24) compreendem Γ. A teoria de base Σ pode ser considerada como o conjunto de Fórmulas (25) à (33). Por fim, os predicados abdutíveis são *receding/3*, *approaching/3*, *splitting/3* e *coalescing/3*.

Dada uma sequência de mapas de profundidade descritos pela Fórmula (34), onde O_1 e O_2 representam objetos no mundo e V_1 e V_2 pontos de vista de onde os mapas foram observados, através de abdução pode-se gerar as Fórmulas (35) e (36) como forma de interpretação dos dados sensoriais que descrevem a cena.

$$\begin{aligned} located(Robot_1, V_1, T_0) \wedge located(Robot_1, V_2, T_1) \wedge \\ DC(i(O_1, V_1, T_0), i(O_2, V_1, T_0)) \wedge DC(i(O_1, V_2, T_1), i(O_2, V_2, T_1)) \wedge \\ (dist(i(O_1, V_1, T_0), i(O_2, V_1, T_0)) > dist(i(O_1, V_2, T_1), i(O_2, V_2, T_1))) \end{aligned} \tag{34}$$

Um robô denominado $Robot_1$ observa dois objetos O_1 e O_2 de dois pontos de vista distintos em instantes de tempo diferentes, de modo que em ambos os momentos os objetos estão desconectados e a distância entre eles era maior no ponto de vista V_1 do que em V_2.

$$\begin{aligned} \exists v \exists t\ located(Robot_1, v, t) \wedge between(v, V_1, V_2) \wedge \\ approaching(i(O_1, v, t), i(O_2, v, t)) \wedge T_0 < t \wedge t < T_1 \end{aligned} \tag{35}$$

A Fórmula (35) é obtida através das Fórmulas (34) e (25). Da mesma forma que a Fórmula (36)

$$ap_getting_closer(O_1, O_2, v, t), \tag{36}$$

é obtida através das Fórmulas (35), (30) e utilizando a premissa de objetos imóveis. Este resultado é uma hipótese sobre um possível estado do mundo que explica os dados sensoriais. Deste ponto, se for considerado que o observador continuou seu movimento após o segundo ponto de vista, é possível deduzir uma previsão futura.

$$\begin{aligned} &\exists w_1 \exists u_1\ located(Robot_1, w_1, u_1) \wedge between(w_1, V_2, V_3) \wedge \\ &\quad EC(i(O_1, w_1, u_1), i(O_2, w_1, u_1)) \wedge T_1 < u_1 \end{aligned} \tag{37}$$

A Fórmula (37) representa que os objetos O_1 e O_2 estarão externamente conectados em um instante de tempo posterior à T_1. Desta hipótese, em conjunto com o Axioma (31), obtem-se que em algum instante de tempo posterior à u_1 os objetos estarão se fundindo.

$$\begin{aligned} &\exists w_2 \exists u_2\ located(Robot_1, w_2, u_2) \wedge \\ &\quad coalescing(i(O_1, w_2, u_2), i(O_2, w_2, u_2)) \wedge u_1 < u_2 \end{aligned} \tag{38}$$

E por fim, através da Fórmula (39) e do Axioma (31) pode-se chegar numa hipótese de oclusão no instante de tempo u_2.

$$occluding(O_1, O_2, w_2, u_2) \tag{39}$$

Santos e Shanahan (2002) é parte de uma sequência de publicações que levam até o presente trabalho, onde são investigadas relações espaciais qualitativas do ponto de vista de um robô móvel. Os trabalhos mais relevantes após este são Santos (2007), descrito na próxima seção, e Soutchanski e Santos (2008) descrito no próximo capítulo.

2.1.4 Raciocínio sobre profundidade e movimento

Em Santos (2007) foi investigado um formalismo para raciocínio espacial sobre profundidade e movimento a partir do ponto de vista de um observador. A representação simbólica dos dados do sistema de visão estéreo do robô leva em consideração uma simplificação dos mapas de profundidade gerados. Considera-se uma faixa horizontal no centro do campo de visão. Cada faixa contém informação de profundidade dos objetos em cena em duas dimensões e este plano de visão constitui a entidade básica do sistema, que é chamado de *Depth Profile Calculus* (DPC – Cálculo sobre Perfil de Profundidade).

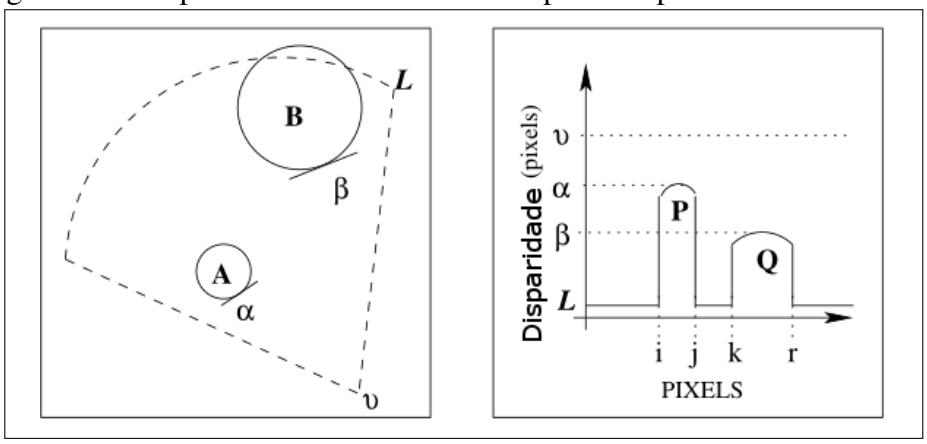
Este trabalho foi baseado no conceito de que a percepção robótica é um processo passivo onde a informação sensorial é assimilada por um robô através de um processo de *abdução*,

alterando sua crença em relação ao ambiente em que o mesmo se encontra (SHANAHAN, 1996).

As entidades básicas do DPC são picos de perfis de profundidade gerados do ponto de vista do observador em relação a objetos que se encontram no seu campo de visão. Cada perfil de profundidade corresponde a uma cena do mundo e é limitado pela direção em que o robô móvel está posicionado, ao ângulo de abertura máximo e ao alcance de profundidade do seu sensor. A área determinada por esse ângulo é chamada de cone de visibilidade. Nesta teoria, por questões de simplicidade considera-se que todos os objetos no mundo são cilindros. Além disso, outros aspectos importantes desta teoria são os instantes no tempo em que as ações são realizadas e as noções de profundidade (disparidade), tamanho e distância angular. Algumas regras de domínio também são determinadas nesta teoria. Assume-se que cada pico presente num perfil de profundidade representa um objeto físico no mundo, com exceção de quando houver uma razão para que se acredite no contrário. Além disso, é considerado que no domínio os objetos não podem desaparecer ou aparecer de um instante para outro (*persistência de objetos*), é necessária uma transição completa entre estados, o que define outra regra de domínio onde cada mudança de localização do robô é devida a um movimento contínuo dos objetos (*continuidade*). Por fim, também é determinado como regra de que os objetos do mundo são substanciais e por este motivo não podem atravessar uns aos outros (*substancialidade)*.

A disparidade de um pico de profundidade é medida através da diferença entre um limite L que limita o alcance do sensor do robô e o pico do sinal. Assim, quanto maior o valor de disparidade, mais próximo o objeto se encontra do observador. O tamanho é medido através do módulo da diferença entre as laterais de cada pico, e a distância angular é o módulo da diferença entre as laterais mais próximas de cada pico. Na Figura 3 estão ilustrados o campo de visão de um robô com dois objetos visíveis, e o seu respectivo perfil de profundidade. Neste exemplo, o tamanho do pico Q é definido pelo módulo de $k - r$ e a distância entre os picos P e Q pelo módulo de $k - j$.

Figura 3 – Campo de visão do observador e perfil de profundidade da cena



Fonte: Autor "adaptado de"Santos, 2007

Tempo é outro importante aspecto da teoria DPC e ele é explorado através da noção de intervalo de tempo I e pelo predicado $meet/2$, onde um par de intervalos $I_1 = \langle t_1, t_2 \rangle\, meet\, I_2 = \langle t_3, t_4 \rangle$ se e somente se $t_2 = t_3$. As funções que definem distância, tamanho e disparidade são respectivamente $dist(x,y,t)$, onde x e y são picos no perfil de profundidade no instante t, $size(x,t)$, onde x é um pico no perfil de profundidade no instante t, $disp(x,t)$, onde x é um pico no perfil de profundidade no instante de tempo t. Através dessas funções são definidas as relações: $DisC(x,y,t)$, onde x está desconectado de y no instante t; $ExtC(x,y,t)$, onde x está externamente conectado a y no instante t; $Co(x,y,t)$, onde x está fundido a y no instante t; $Further_Than(x,y,t)$ ($FT(x,y,t)$), onde x está mais longe do observador do que y no instante t; $Closer_Than(x,y,t)$ ($CT(x,y,t)$), onde x está mais perto do observador do que y no instante t; $Depth_Equal(x,y,t)$ ($DEq(x,y,t)$), onde x está tão longe do observador quanto y no instante t; $Larger_Than(x,y,t)$ ($LT(x,y,t)$), onde x é maior do que y no instante t; $Smaller_Than(x,y,t)$ ($ST(x,y,t)$), onde x é menor que y no instante t; $Size_Equal(x,y,t)$ ($SEq(x,y,t)$), onde x possui o mesmo tamanho de y no instante t.

O DPC assume que para qualquer instante de tempo ambas as bordas do campo de visão do robô ($LeftBorder$ e $RightBorder$) são desconexas de modo que para todo pico p no instante t as seguintes sentenças são verdadeiras: $CT(LeftBorder,p,t)$, $FT(p,RightBorder,t)$, $LT(LeftBorder,p,t)$ e $ST(p,RightBorder,t)$.

Os Axiomas (40), (41) e (42) definem as funções de distância. São necessários dois limiares δ_1 e δ_2 de modo que $\delta_2 > 0$ e $\delta_1 < \delta_2$. Estes limiares são determinados pelo domínio. Levando-se em consideração o domínio de um robô móvel estes limiares poderiam ter o valor da largura do robô de modo que na possibilidade de avaliação da função $ExtC/3$, o robô teria a percepção de que seu caminho está obstruído e por este motivo são necessárias ações de contorno para que o mesmo continue a se mover.

$$Disc(x,y,t) \leftrightarrow (dist(x,y,t) > \delta_2); \tag{40}$$

$$ExtC(x,y,t) \leftrightarrow (\delta_1 < dist(x,y,t)) \wedge (dist(x,y,t) \leq \delta_2); \tag{41}$$

$$Co(x,y,t) \leftrightarrow (dist(x,y,t) \leq \delta_1). \tag{42}$$

Os Axiomas (43), (44) e (45) definem as funções de disparidade. É necessário um valor μ para representar um desvio médio na aferição de disparidade do sensor.

$$FT(x,y,t) \leftrightarrow (disp(x,t) < disp(y,t)) \wedge (|disp(x,t) - disp(y,t)| \geq \mu); \tag{43}$$

$$CT(x,y,t) \leftrightarrow (disp(x,t) > disp(y,t)) \wedge (|disp(x,t) - disp(y,t)| \geq \mu); \tag{44}$$

$$DEq(x,y,t) \leftrightarrow (|disp(x,t) - disp(y,t)| < \mu). \tag{45}$$

Os Axiomas (46), (47) e (48) definem as funções de tamanho. É necessário um valor σ para representar um erro de aferição de tamanho do sensor.

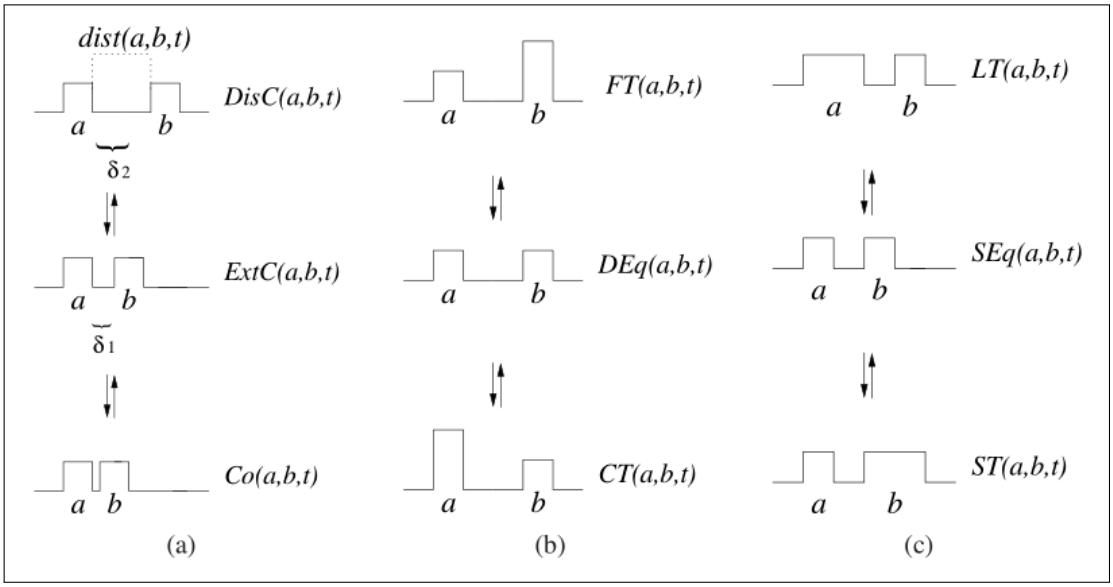
$$LT(x,y,t) \leftrightarrow (size(x,t) > size(y,t)) \wedge (|size(x,t) - size(y,t)| \geq \sigma); \tag{46}$$

$$ST(x,y,t) \leftrightarrow (size(x,t) < size(y,t)) \wedge (|size(x,t) - size(y,t)| \geq \sigma); \tag{47}$$

$$SEq(x,y,t) \leftrightarrow (|size(x,t) - size(y,t)| < \sigma). \tag{48}$$

Na Figura 4 pode-se observar todas as relações de distância, disparidade e tamanho e notar que o conjunto de relações do DPC são mutuamente exclusivas e disjuntas par a par.

Figura 4 – Diagrama de transições para as relações de distância, disparidade e tamanho



Fonte: Santos, 2007

O conjunto total de relações do DPC é formado pela composição das relações $DisC/3$, $ExtC/3$ e $Co/3$, totalizando 27 relações básicas que possuem a forma:

$$\begin{aligned}
&(DPC1 - DPC27) R_n^{DPC}(x,y,t) \leftrightarrow R_i^{PDist}(x,y,t) \wedge R_j^{PDepth}(x,y,t) \wedge R_k^{PSize}(x,y,t),\\
&onde, R_i^{PDist}(x,y,t) \in \{Disc(x,y,t), ExtC(x,y,t), Co(x,y,t)\},\\
&R_j^{PDepth}(x,y,t) \in \{DEq(x,y,t), FT(x,y,t), CT(x,y,t)\},\\
&R_k^{PSize}(x,y,t) \in \{SEq(x,y,t), LT(x,y,t), ST(x,y,t)\},\\
&e\, n \in 1,2,3,\ldots,27.
\end{aligned}$$

Através das relações básicas apresentadas, também foram definidas em Santos (2007) relações dinâmicas de transição em atributos de picos nos perfis de profundidade. Para isso, foi definido que um predicado $In_Sight(p,t)$ é verdadeiro se e somente se o pico $p = p_o(a,t)$, que representa o pico p do objeto a no instante de tempo t, não é nulo em tempo t. Sempre que cada conjunto de relações básicas $PDist$, $PDepth$ e $PSize$ possuir alguma relação verdadeira para um pico p num instante de tempo t, então o objeto que p representa está dentro do campo de visão do robô. Da mesma forma que se $In_Sight(p, I)$ é verdadeiro, significa que para todo instante de tempo t pertencente ao intervalo de tempo I, $In_Sight(p,t)$ também é verdadeiro.

A relação de transição dinâmica para um pico no perfil de profundidade $Extending(p,t)$ representa que a disparidade do pico p está aumentando no instante de tempo t, e o objeto por ele representado está sempre visível num curto intervalo de tempo que contenha t. Assume-se um limiar de disparidade de valor $\epsilon = (L + dL)$, onde L é o limite de alcance do campo de visão do observador e dL é uma curta distância em respeito a L. Este limiar é utilizado para diferenciar as relações de aparição ($Appearing/2$), de extensão ($Extending/2$), e as relações de desaparecimento ($Vanishing/2$) e de diminuição ($Shrinking/2$). Formalmente a relação $Extending/2$ é definida pelo Axioma (49), onde um pico p está crescendo em tamanho num instante t se e somente se sua disparidade aumentar de um perfil de profundidade em tempo anterior t_1 para outro perfil em t_2, formando um intervalo curto de tempo $I = \langle t_1,t_2 \rangle$ que contém t, e as disparidades em cada instante de tempo permanecem maiores do que o limiar ϵ.

$$Extending(p,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge (disp(p,t_1) < disp(p,t_2)) \wedge \\ (disp(p,t_1) \geq \epsilon) \wedge (disp(p,t_2) \geq \epsilon). \tag{49}$$

A relação $Shrinking(p,t)$ define que a disparidade do pico p está diminuindo no instante de tempo t. Além disso, da mesma forma que em $Extending/2$, p permanece sempre visível num intervalo de tempo curto que contenha t. Formalmente esta relação é definida pelo Axioma (50). A diferença para o Axioma (49) é que nesta relação a disparidade do pico p diminuiu no intervalo de tempo.

$$Shrinking(p,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge (disp(p,t_1) > disp(p,t_2)) \wedge \\ (disp(p,t_1) \geq \epsilon) \wedge (disp(p,t_2) \geq \epsilon). \tag{50}$$

A relação $Vanishing(p,t)$ define que a disparidade do pico p está diminuindo no instante t para um ponto em que ele não pode mais ser percebido. Formalmente esta relação é definida pelo Axioma (51).

$$Vanishing(p,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge (disp(p,t_1) > disp(p,t_2)) \wedge \\ (disp(p,t_2) < \epsilon). \tag{51}$$

A relação $Appearing(p,t)$ representa que um pico p está aparecendo no fundo de um perfil de profundidade, complementando a relação $Vanishing/2$. Formalmente esta relação é definida pelo Axioma (52).

$$Appearing(p,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge (disp(p,t_1) < disp(p,t_2)) \wedge (disp(p,t_1) < \epsilon). \quad (52)$$

A relação $Peak_approaching_DisC(p,q,t)$ representa que os picos p e q estão se aproximando mas permanecem desconectados no instante de tempo t. Formalmente esta relação é definida pelo Axioma (53).

$$Peak_approaching_DisC(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge DisC(p,q,t_1) \wedge Disc(p,q,t_2) \wedge (disp(p,q,t_2) < disp(p,q,t_1)). \quad (53)$$

A relação $Peak_approaching_ExtC(p,q,t)$ representa que os picos p e q estão se aproximando e se tornam externamente conectados no instante de tempo t. Formalmente esta relação é definida pelo Axioma (54).

$$Peak_approaching_ExtC(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge (DisC(p,q,t_1) \vee ExtC(p,q,t_1)) \wedge ExtC(p,q,t_2) \wedge (disp(p,q,t_2) < disp(p,q,t_1)). \quad (54)$$

A relação $Peak_receding_Disc(p,q,t)$ representa que os picos p e q que já estavam desconectados estão se afastando no instante de tempo t. Formalmente esta relação é definida pelo Axioma (55).

$$Peak_receding_DisC(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge DisC(p,q,t_1) \wedge Disc(p,q,t_2) \wedge (disp(p,q,t_2) > disp(p,q,t_1)). \quad (55)$$

A relação $Peak_receding_ExtC(p,q,t)$ representa que os picos p e q que estavam externamente conectados estão se afastando no instante de tempo t. Formalmente esta relação é definida pelo Axioma (56).

$$Peak_receding_ExtC(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge ExtC(p,q,t_1) \wedge (DisC(p,q,t_2) \vee ExtC(p,q,t_2)) \wedge (disp(p,q,t_2) > disp(p,q,t_1)). \quad (56)$$

A relação $Peak_coalescing(p,q,t)$ representa que os picos p e q estão se unificando no instante de tempo t. Formalmente esta relação é definida pelo Axioma (57).

$$Peak_coalescing(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge ExtC(p,q,t_1) \wedge Co(p,q,t_2). \quad (57)$$

A relação $Going_out_sight(p,q,t)$ representa o caso de um dos picos p e q estar desaparecendo do campo de visão do observador num instante de tempo t, mas não se sabe exatamente qual dos picos. Essa incerteza é devido ao evento estar diretamente conectado a uma interpretação de alto nível do contexto em que a situação de desaparecimento ocorre. Formalmente esta relação é definida pelo Axioma (58).

$$\begin{aligned} Going_out_sight(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge Peak_coalescing(p,q,t_1) \wedge \\ [(In_Sight(p,t_1) \wedge \neg In_Sight(p,t_2)) \vee \\ (In_Sight(q,t_1) \wedge \neg In_Sight(q,t_2))]. \end{aligned} \quad (58)$$

A relação $Peak_Splitting(p,q,t)$ define que um pico se separa em dois picos distintos p e q num instante de tempo t. Formalmente esta relação é definida pelo Axioma (59).

$$Peak_Splitting(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge Co(p,q,t_1) \wedge ExtC(p,q,t_2). \quad (59)$$

A relação $Single_peak_static(q,t)$ representa que um pico q não sofreu alteração de tamanho e disparidade num instante de tempo t. Formalmente esta relação é definida pelo Axioma (60).

$$\begin{aligned} Single_peak_static(q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge \\ In_Sight(p,[t_1,t_2]) \wedge (size(p,t_1) = size(p,t_2)) \wedge \\ (disp(p,t_1) = disp(p,t_2)). \end{aligned} \quad (60)$$

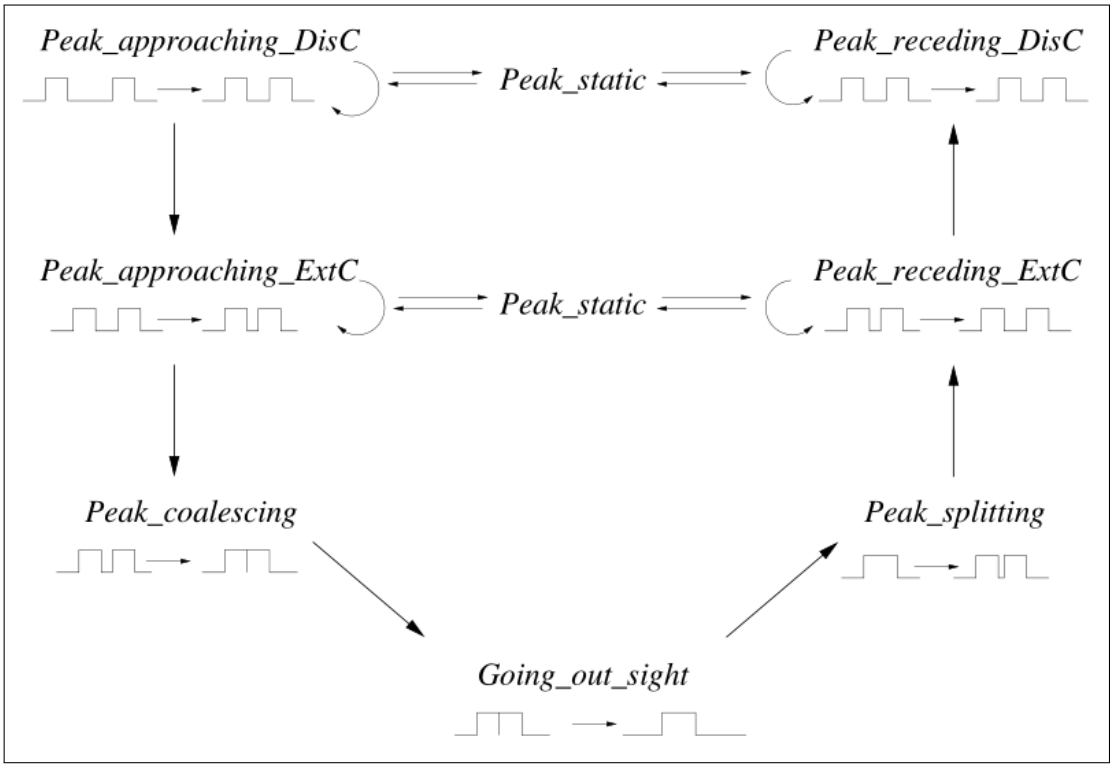
A relação $Peak_static(p,q,t)$ representa que os picos p e q não sofreram alteração de distância angular entre eles no instante de tempo t. Formalmente esta relação é definida pelo Axioma (61).

$$\begin{aligned} Peak_static(p,q,t) \leftrightarrow \exists t_1 \exists t_2 (t_1 < t) \wedge (t < t_2) \wedge \\ In_Sight(p,[t_1,t_2]) \wedge In_Sight(q,[t_1,t_2]) \wedge (disp(p,q,t_1) = disp(p,q,t_2)). \end{aligned} \quad (61)$$

A partir das relações dinâmicas de transição apresentadas é possível construir um diagrama conceitual de vizinhança conforme Figuras (5) e (6). Entre cada conexão ligando um par de relações seria possível adicionar as relações $Peak_static/3$ e $Single_peak_static/2$ pois tais relações sempre podem existir, em qualquer instante.

Cada conexão entre as relações das Figuras (5) e (6) pode ser considerada como um teorema em separado. A prova de cada conexão está publicada em (SOUTCHANSKI; SANTOS, 2008). A teoria diz que cada conexão no diagrama conceitual de vizinhança representa a

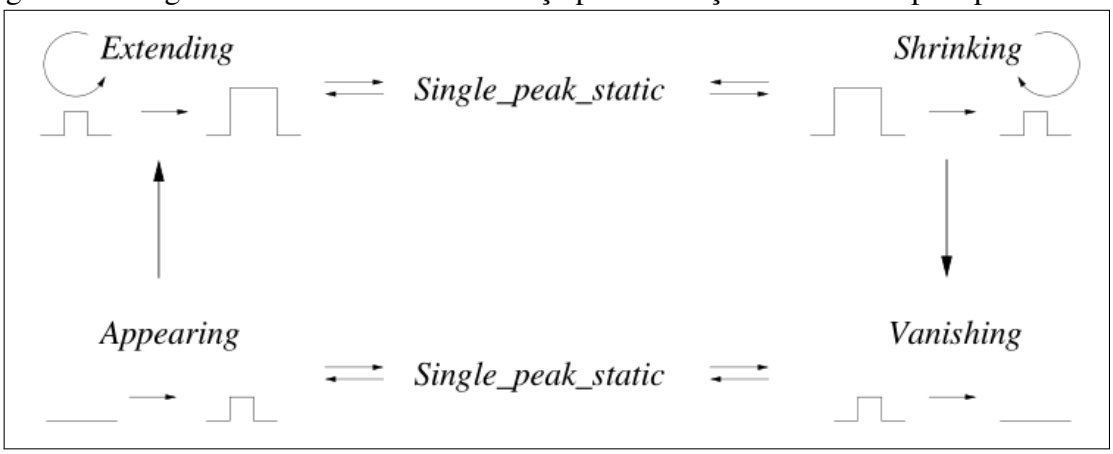
Figura 5 – Diagrama conceitual de vizinhança para transições em picos de profundidade



Fonte: Santos, 2007

menor disjunção que pode descrever cada transição contínua de todos os modelos de axiomas das relações (49) a (61). O conceito de transição contínua significa que para qualquer ponto no tempo $u = (t,t')$, em que uma determinada relação espacial φ exista para os instantes t e t', não existe outra relação qualitativa desta teoria que qualifique a relação em p e q (SANTOS, 2007).

Figura 6 – Diagrama conceitual de vizinhança para transições dinâmicas para pico único



Fonte: Santos, 2007

Devido à natureza do observador ser um robô móvel, a informação que ele adquire sobre o mundo é relacionada ao seu ponto de vista. Por este motivo, as conclusões a respeito do movimento de objetos no mundo que se pode chegar são hipóteses explanatórias e não deduções sobre fatos. Ao assumir que as consequências dos axiomas são verdadeiras, é possível

gerar hipóteses sobre o domínio através de um processo de abdução. As chamadas hipóteses objeto-observador Santos (2007) são classes de relações espaciais dependentes do tempo nas quais as relações previamente definidas estão relacionadas. Estas classes podem ser separadas exclusivamente pela informação presente nos perfis de profundidade.

A primeira classe conecta as relações de pico simples $Extending/2$, $Shrinking/2$, $Appearing/2$ e $Vanishing/2$ com os seguintes predicados para objetos:

a) $Being_approached_by(a,t)$, que representa que um objeto a está se aproximando do observador no instante de tempo t;

b) $Moving_away_from(a,t)$, que representa que um objeto a está se afastando do observador no instante de tempo t;

c) $Approaching(a,t)$, que representa que o observador está se aproximando de um objeto a no instante de tempo t;

d) $Receding(a,t)$, que representa que o observador está se afastando do objeto a no instante de tempo t;

e) $Expanding(a,t)$, que representa que o objeto a está expandindo no instante de tempo t;

f) $Contracting(a,t)$, que representa que o objeto a está se contraindo no instante de tempo t.

A formalização dessa conexão pode ser vista nos Axiomas (62) e (63).

$$(Extending(p,t) \vee Appearing(p,t)) \leftarrow p = p_o(a,t) \wedge \\ (Being_approached_by(a,t) \vee Approaching(a,t) \vee Expanding(a,t)); \quad (62)$$

$$(Appearing(p,t) \vee Vanishing(p,t)) \leftarrow p = p_o(a,t) \wedge \\ (Moving_away_from(a,t) \vee Receding(a,t) \vee Contracting(a,t)). \quad (63)$$

A segunda classe conecta as relações de picos $Peak_approaching_DisC/3$, $Peak_approaching_ExtC/3$, $Peak_receding_DisC/3$, $Peak_receding_ExtC/3$ com as seguintes relações de objetos:

a) $Getting_closer(a,b,t)$, que representa que os objetos a e b estão se aproximando no instante de tempo t;

b) $Getting_further(a,b,t)$, que representa que os objetos a e b estão se afastando no instante de tempo t;

c) $Circling_{<}(a,b,t)$, que representa que a trajetória realizada pelo observador é um arco ao redor dos objetos a e b no instante de tempo t de tal forma que a distância aparente de suas imagens está diminuindo;

d) $Circling_{>}(a,b,t)$, que representa que a trajetória realizada pelo observador é um arco ao redor dos objetos a e b no instante de tempo t de tal forma que a distância aparente de suas imagens está aumentando;

e) $Moving_backward(a,b,t)$, que representa que o observador está se movendo na direção contrária aos objetos a e b no instante de tempo t;

f) $Moving_forward(a,b,t)$, que representa que o observador está se movendo em direção aos objetos a e b no instante de tempo t;

g) $Obj_moving_backward(a,b,t)$, que representa que os objetos estão se movendo na direção contrária do observador no instante de tempo t;

h) $Obj_moving_forward(a,b,t)$, que representa que os objetos estão se movendo em direção ao observador no instante de tempo t.

A formalização dessa conexão pode ser vista nos Axiomas (64) e (65).

$$\begin{aligned}&(Peak_approaching_DisC(p,q,t) \vee Peak_approaching_ExtC(p,q,t)) \leftarrow \\ &\qquad p = p_o(a,t) \wedge q = p_o(b,t) \wedge \\ &\qquad (Getting_closer(a,b,t) \vee Circling_{<}(a,b,t) \vee \\ &\qquad Moving_backward(a,b,t) \vee Obj_moving_backward(a,b,t));\end{aligned} \tag{64}$$

$$\begin{aligned}&(Peak_receding_DisC(p,q,t) \vee Peak_receding_ExtC(p,q,t)) \leftarrow \\ &\qquad p = p_o(a,t) \wedge q = p_o(b,t) \wedge \\ &\qquad (Getting_further(a,b,t) \vee Circling_{>}(a,b,t) \vee \\ &\qquad Moving_forward(a,b,t) \vee Obj_moving_forward(a,b,t)).\end{aligned} \tag{65}$$

Vale notar que podem ocorrer múltiplas hipóteses devido à natureza do problema de interpretação de observações. Por exemplo, na hipótese da relação $Peak_approaching_DisC(p,q,t)$ ser verdadeira, além das hipóteses objeto-observador presentes no lado direito do Axioma (64), mas também pela hipótese de ambos os objetos a e b representados pelos picos p e q estarem se estendendo conforme Axioma (62).

Ainda nesta mesma classe, ao utilizar as bordas do campo de visão ($RightBorder$ e $LeftBorder$) como argumentos das relações de picos $Peak_approaching_DisC/3$, $Peak_approaching_ExtC/3$, $Peak_receding_DisC/3$, $Peak_receding_ExtC/3$, são obtidas novas hipóteses do ponto de vista do observador. $Turning_right(a,t)$, que representa

que o observador está virando à direita no instante de tempo t enquanto observa o objeto a; $Turning_left(a,t)$, que representa que o observador está virando à esquerda no instante de tempo t enquanto observa o objeto a; $Moving_right(a,t)$, que representa que o objeto a está se movendo para a direita do observador no instante de tempo t; $Moving_left(a,t)$, que representa que o objeto a está se movendo para a esquerda do observador no instante de tempo t. Essas hipóteses são formalizadas pelos Axiomas (66), (67), (68) e (69).

$$(Peak_approaching_DisC(LeftBorder,q,t) \vee Peak_approaching_ExtC(LeftBorder,q,t)) \leftarrow q = p_o(b,t) \wedge (Turning_right(b,t) \vee Moving_Left(b,t)). \quad (66)$$

$$(Peak_approaching_DisC(p,RightBorder,t) \vee Peak_approaching_ExtC(p,RightBorder,t)) \leftarrow p = p_o(a,t) \wedge (Turning_Left(a,t) \vee Moving_right(a,t)). \quad (67)$$

$$(Peak_receding_DisC(LeftBorder,q,t) \vee Peak_receding_ExtC(LeftBorder,q,t)) \leftarrow q = p_o(b,t) \wedge (Turning_left(b,t) \vee Moving_right(b,t)). \quad (68)$$

$$(Peak_receding_DisC(p,RightBorder,t) \vee Peak_receding_ExtC(p,RightBorder,t)) \leftarrow p = p_o(a,t) \wedge (Turning_right(a,t) \vee Moving_left(a,t)). \quad (69)$$

Os Axiomas (64) a (69) não foram mais precisamente definidos por não terem sido consideradas outras fontes de conhecimento, como por exemplo as ações tomadas por cada objeto.

A terceira classe conecta as relações de picos $Peak_coalescing/3$ e $Going_out_sight/3$ com as seguintes hipóteses sobre objetos no mundo: $Occluding(a,b,t)$, que representa que o objeto a oclui o objeto b no instante de tempo t; $Touching(a,b,t)$, que representa que o objeto a está tocando o objeto b no instante de tempo t e ambos os objetos permanecem visíveis ao observador; $Totally_occluding(a,b,t)$, que representa que o objeto a oclui totalmente o objeto b do ponto de vista do observador no instante de tempo t; $Observer_just_passed_right(a,t)$, que representa que o observador acabou de passar pelo lado direito do objeto a; $Obj_just_passed_right(a,t)$, que representa que o objeto acabou de passar pela direita do observador; $Observer_just_passed_left(a,t)$, que representa que o observador acabou de passar

pelo lado esquerdo do objeto a; $Obj_just_passed_left(a,t)$, que representa que o objeto acabou de passar pela esquerda do observador. A formalização destas hipóteses pode ser notada nos Axiomas (70) a (76).

$$Peak_coalescing(p,q,t) \wedge CT(p,q,t) \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Occluding(a,b,t). \quad (70)$$

$$Peak_coalescing(p,q,t) \wedge FT(p,q,t) \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Occluding(a,b,t). \quad (71)$$

$$Peak_coalescing(p,q,t) \wedge DEq(p,q,t) \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Touching(a,b,t). \quad (72)$$

$$Going_out_sight(p,q,t) \wedge CT(p,q,t_1) \wedge t_1 < t \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Totally_occluding(a,b,t). \quad (73)$$

$$Going_out_sight(p,q,t) \wedge FT(p,q,t_1) \wedge t_1 < t \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Totally_occluding(a,b,t). \quad (74)$$

$$Going_out_sight(RightBorder,q,t) \leftarrow q = p_o(b,t) \wedge (Observer_just_passed_right(b,t) \vee Obj_just_passed_right(b,t)). \quad (75)$$

$$Going_out_sight(p,LeftBorder,t) \leftarrow p = p_o(a,t) \wedge (Observer_just_passed_left(b,t) \vee Obj_just_passed_left(b,t)). \quad (76)$$

A quarta classe conecta a relação $Peak_Splitting/3$ com a hipótese $Appearing_behind(a,b,t)$, que representa que o objeto a está aparecendo atrás do objeto b no instante de tempo t. A formalização dessa hipótese está descrita nos Axiomas (77) e (78).

$$Peak_splitting(p,q,t) \wedge CT(p,q,t) \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Appearing_behind(b,a,t). \quad (77)$$

$$Peak_splitting(p,q,t) \wedge FT(p,q,t) \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge$$

$$Appearing_behind(a,b,t). \tag{78}$$

A quinta e última classe conecta as relações $Peak_static/3$ e $Single_peak_static/2$ com as hipóteses $Object_static(a,b,t)$, que representa que os objetos a e b não estão se movendo no instante de tempo t em relação ao ponto de vista do observador, e $Single_object_static(a,t)$, que representa que o objeto a está estático no instante de tempo t em relação ao ponto de vista do observador. A formalização destas hipóteses pode ser vista nos Axiomas (79) e (80).

$$Peak_static(p,q,t) \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Object_static(a,b,t). \tag{79}$$

$$Single_peak_static(p,t) \leftarrow p = p_o(a,t) \wedge q = p_o(b,t) \wedge Single_object_static(a,t). \tag{80}$$

Com base nas formalizações apresentadas, foi definido em Santos (2007) que o processo de explicação dos acontecimentos do mundo é realizado inferindo o lado esquerdo das fórmulas (49) a (61) dadas observações do mundo em função de transições de picos em perfis de profundidade de campo que fazem correspondência ao lado direito das mesmas. Com as transições inferidas, através de um processo de abdução são levantadas hipóteses sobre os objetos do mundo de acordo com as fórmulas (62) a (80). Em contrapartida, tudo aquilo que não se pode ser explicado por abdução, e não é uma consequência deste conjunto de axiomas, é considerado falso.

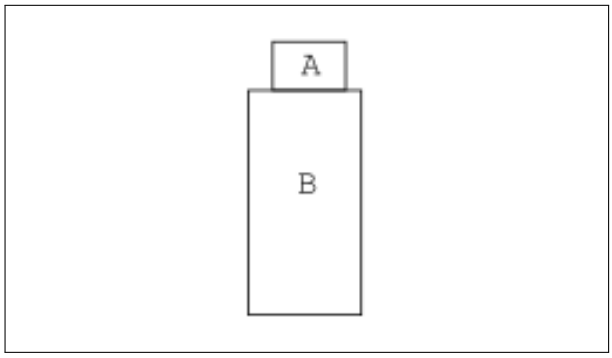
Santos (2007) definiu as relações espaciais qualitativas que foram estudadas neste trabalho, porém sem considerar uma linguagem de ações que possuem influência no mundo, que é o foco do Capítulo 3.

2.2 TRABALHOS RECENTES

Takahashi (2012) baseou-se no RCC8, mas ao contrário de Santos (2007) foram utilizadas imagens adquiridas de dois pontos de vista distintos num mesmo instante e imagens de vídeo para inferir relações espaciais em três dimensões. A motivação da investigação é de que em algumas situações as relações do RCC8 gerariam múltiplas hipóteses que poderiam ser melhor definidas caso existisse um segundo ponto de vista. Um exemplo seria a Figura (7) onde não é possível definir se o objeto A está externamente conectado ao objeto B, ou se o objeto B oclui parcialmente o objeto A.

Porém, ao analisar uma sequência de imagens conforme Figura (8), é possível verificar que no instante de tempo t a relação correta seria a de que o objeto B oclui parcialmente o objeto A. No entanto, caso o objeto B possua a forma de um tubo de forma que o objeto A passe por dentro dele, ainda não seria possível concluir que o objeto B está mais próximo do observador do que o objeto A. Por este motivo foi estudado um formalismo baseado em dois

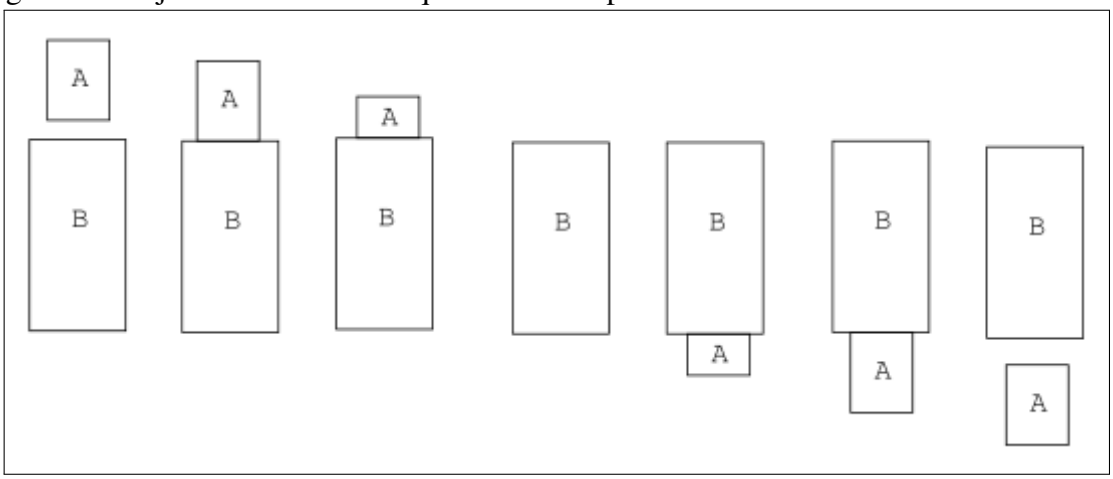
Figura 7 – Objetos A e B num determinado instante t



Fonte: Takahashi, 2012

observadores. Devido ao presente trabalho se concentrar na investigação de uma teoria de um único observador, não nos aprofundaremos nesta teoria.

Figura 8 – Objetos A e B numa sequência de tempo



Fonte: Takahashi, 2012

Em Albath et al. (2010) foi desenvolvida uma extensão do RCC8 chamada RCC3D. Nesta teoria foram estudadas imagens da área de anatomia, de um ponto de vista ortogonal de um objeto de três dimensões. A conclusão é de que não havia informação suficiente para aplicar RCC8 apenas com a informação presente nas imagens e por este motivo determinou-se que a informação de extensão do objeto em cena é conhecida *a priori*, motivando o desenvolvimento do RCC3D. As relações do RCC8 foram redefinidas no RCC3D como:

a) $P(A,A)$, que representa que A é parte dele mesmo;

b) $P_c(A,B)$, que representa o inverso de A é parte de B, ou seja, B é parte de A;

c) $P(A,B) \wedge P_c(A,B) \rightarrow A = B$, que representa que se A é parte de B e o inverso é verdadeiro, então $A = B$;

d) $P(A,B) \wedge P(B,C) \rightarrow P(A,C)$, que representa que se A é parte de B e B é parte de C, então A é parte de C;

e) $C(A,A)$, onde A está conectado a ele mesmo;

f) $C(A,B) \leftrightarrow C(B,A)$, onde A está conectado à B se e somente se B estiver conectado à A;

g) $PP(A,B) \leftrightarrow P(A,B) \wedge \neg P_c(A,B)$, onde A é parte própria de B se e somente se A for parte de B e B não for parte de A;

h) $PO(A,B) \leftrightarrow (A^o \cap B^o \neq \varphi) \wedge \neg P(A,B) \wedge \neg P_c(A,B)$, que representa que A oclui propriamente B se e somente se a intersecção entre o interior de A (A^o) e interior de B (B^o) não for um conjunto vazio e A não for parte de B e B não for parte de A;

i) $EQ(A,B) \equiv A = B$, que representa que A é igual a B;

j) $EQ(A,B) \equiv C(A,B) \wedge (A^o \cap B^o = \varphi)$, que representa que se A está externamente conectado a B, equivale dizer que A está conectado a B e a intersecção entre os interiores de A e B é um conjunto vazio;

k) $DC(A,B) \equiv \neg C(A,B)$, que representa que se A está desconectado de B equivale dizer que A não está conectado a B;

l) $TPP(A,B) \equiv PP(A,B) \wedge \exists (C \neq \varphi)[EC(C,A^o) \wedge EC(C,B^o)]$, que representa que se A é parte própria tangencial de B então equivale dizer que A é parte própria de B e existe uma região C não vazia que está externamente conectada ao interior de A e ao interior de B;

m) $NTPP(A,B) \equiv PP(A,B) \wedge \neg \exists C[EC(C,A) \wedge EC(C,B)]$, que representa que se A é parte própria não tangencial de B então equivale dizer que A é parte própria de B e não existe uma região C externamente conectada à A e B;

n) $TPP_c(A,B) \equiv PP_c(A,B) \wedge \exists (C \neq \varphi)[EC(C,A^o) \wedge EC(C,B^o)]$, que representa que se B é parte própria tangencial de A então equivale dizer que B é parte própria de A e existe uma região C não vazia que está externamente conectada ao interior de A e ao interior de B;

o) $NTPP_c(A,B) \equiv PP_c(A,B) \wedge \neg \exists C[EC(C,A) \wedge EC(C,B)]$, que representa que se B é parte própria não tangencial de A então equivale dizer que B é parte própria de A e não existe uma região C externamente conectada à A e B.

Em Albath, Leopold e Sabharwal (2010) foi desenvolvido um sistema com interface visual chamada *VRCC-3D* capaz de inferir relações espaciais do RCC3D e permitir que um usuário verificasse inconsistências. Em Sabharwal, Leopold e Eloe (2011) verificou-se que o modelo apresentava resultados ambíguos em determinadas situações, o que levou à adição de um componente de profundidade, aumentando o número de relações compostas de 13 para 34, renomeando a teoria para *VRCC-3D+*.Os tipos possíveis de oclusão possuem a forma *xObsy*,

onde *Obs* faz referência à intersecção entre as projeções dos objetos. O prefixo x pode ser n (não oclusão), p (oclusão parcial), e (oclusão equivalente), c (oclusão completa). O sufixo y pode ser um espaço em branco que indicaria que A está na frente de B, pode ser $_e$, que representa que ambos os objetos estão equidistantes, ou $_c$, que indicaria que B está na frente de A. Porém, não há uma definição de como o arcabouço interpreta as imagens do ponto de vista do observador.

Outro trabalho que também promove raciocínio sobre ações e mudanças que ocorrem no mundo é Cabalar e Santos (2011), em que foi formalizada uma solução para o problema do *Fisherman's Folly Puzzle*. Foi a primeira formalização lógica para um cenário composto por cordas e objetos rígidos. Além disso, foi apresentada uma noção de conectividade através de um fluente em um formalismo sobre ações e definido um novo formalismo do Cálculo de Situações em Lógica de Equilíbrio Pearce e Valverde (2006) gerando uma solução rigorosa para o problema do quadro (*Frame Problem*), em lógica de primeira ordem.

Ainda dentro da área de raciocínio espacial temporal qualitativo (QSTR), mas com um propósito um pouco diferente, foi definido em Ligozat e Santos (2015) uma representação para raciocínio espacial para múltiplos pontos de vista. O formalismo denominado (*IOC*) estuda o compartilhamento de conhecimento sobre posicionamento e oclusão de objetos entre agentes.

No próximo capítulo será tratada em detalhes a teoria de Soutchanski e Santos (2008), que foi utilizada como base para a implementação e desenvolvimento do presente trabalho.

3 REVISÃO TEÓRICA

A base deste trabalho é a teoria proposta por Soutchanski e Santos (2008) e a estrutura do Cálculo de Situações (REITER, 2001). Neste capítulo ambos os assuntos serão detalhadamente descritos.

3.1 CÁLCULO DE SITUAÇÕES

O Cálculo de Situações (CS) é um formalismo lógico de segunda ordem para representação e raciocínio em domínios dinâmicos. Basicamente a linguagem se baseia em *ações, situações, fluentes* e *objetos*, mas pode ser utilizada em problemas complexos como ações estocásticas, concorrência de ações, tempo contínuo entre outros (REITER, 2001).

A base da linguagem é formada pelo símbolo S_0 que representa a situação inicial e o termo *do(a,s)* que representa a sequência formada após adicionar a ação a na situação s. Além disso, o predicado binário $\subset$ define uma relação de ordem entre situações, onde $s \subset s'$ significa que s é uma subsequência de s'. Existe também o predicado binário *Poss(a,s)* que define a possibilidade de execução da ação a na situação s. Por convenção nas próximas fórmulas lógicas os nomes de variáveis serão escritos em letra minúscula e constantes serão definidas com a primeira letra maiúscula. Os predicados e funções podem ser do tipo:

a) $(action \cup object)^n$, que são predicados binários para relações independentes de situação como por exemplo *humano(Oliveira)* e *addnumero(n)*;

b) $(action \cup object)^n \rightarrow object$, que são funções independentes de situação como por exemplo *sqrt(x)* e *height(MtEverest)*, que resultam num valor;

c) $(action \cup object)^n \rightarrow action$, que são *funções de ação* independentes de situação como *pegar(a)* ou *move(a,b)*, que resultam numa ação;

d) $(action \cup object)^n \times situation$, que são *fluentes relacionais* binários dependentes de uma situação como *ontable(x,s)* e *husband(Mary,John,s)*;

e) $(action \cup object)^n \rightarrow action \cap object$, que são *fluentes funcionais* dependentes de uma situação e que resultam num valor como *age(Mary,s)* ou *primeMinister(Italy,s)*.

Situações são sequências finitas de ações. Duas ou mais situações podem apresentar os mesmos valores-verdade para todos os fluentes e mesmo assim serem diferentes pois o que define a igualdade entre duas situações é o seu histórico de ações. Por este motivo usa-se o termo *situação* e não o termo *estado*, que define apenas valores em determinado momento no tempo, independentemente de seu histórico (REITER, 2001). Outro termo presente no cálculo de situações são as *situações executáveis*, que representam situações em que todas as suas ações podem ser executadas uma após a outra, em sequência.

Ações são termos de primeira ordem constituídos por um símbolo funcional e seus argumentos, sendo que o último argumento geralmente é um representante de tempo (t) que determina o instante em que a ação ocorre. É considerado que a duração do início e do fim de execução de tais ações é zero e por este motivo são utilizados predicados do tipo $startAction(x_1, \ldots, x_n, t)$ e $endAction(x_1, \ldots, x_n, t)$ para determinar o início e fim dos acontecimentos do mundo (REITER, 2001).

Uma *teoria básica de ações* (BAT) é formada por um conjunto de axiomas escritos em ST contendo cinco classes de axiomas: D_{ap}, D_{ss}, D_{una}, D_{S_0} e Σ.

Axiomas de precondição (D_{ap}) são axiomas que determinam a possibilidade de execução de uma ação em uma determinada situação. Cada ação $A(\vec{x})$ possível no mundo possui um axioma da forma

$$Poss(A(\vec{x}), s) \equiv \Pi_A(\vec{x}, s),$$

em que $\Pi_A(\vec{x}, s)$ representa a conjunção de todas as condições necessárias para a execução de A. Essa classe de axiomas é uma solução para o problema de qualificação (*Qualification Problem*) (REITER, 2001). Historicamente ao se axiomatizar ações em domínios dinâmicos sempre existiu um problema para qualificar todas as relações que possibilitam a execução de uma determinada ação, devido existir diversas circunstâncias externas e não inerentes ao domínio que está sendo tratado que precisariam estar explicitamente axiomatizadas. A solução encontrada e descrita por Reiter (2001) envolve a separação destas qualificações em um grupo de maior importância e outro de menor importância. O grupo de maior importância contém as condições que serão efetivamente axiomatizadas pois estão diretamente relacionadas ao domínio. O grupo de menor importância contém condições que possuem alguma possibilidade de ocorrer mas não possuem relação direta com o domínio de conhecimento que se está trabalhando, ou que não se pode saber se a condição é verdadeira ou falsa. Estas condições pertencentes ao grupo de menor importância são conscientemente ignoradas. Segue um exemplo adaptado de (REITER, 2001):

$$\begin{aligned} Poss(pegar(x, s)) \leftrightarrow & [(\forall z) \neg segurando(robo, z, s)] \wedge \neg pesado(x) \wedge perto(robo, x, s) \\ & \wedge \neg coladoAoChao(x, s) \wedge \neg atropeladoCaminhao(robo, s). \end{aligned} \quad (81)$$

No Axioma (81) estão contidas as condições necessárias para que um robô consiga pegar um objeto x na situação s. As condições $\neg coladoAoChao(x, s)$ e $\neg atropeladoCaminhao(r, s)$ exercem influência sobre a possibilidade de se pegar x porém não há como determinar ao certo se são condições verdadeiras ou não em s. Por este motivo, o axioma de precondição assume a forma

$$Poss(pegar(x, s)) \leftrightarrow [(\forall z) \neg segurando(robo, z, s)] \wedge \neg pesado(x) \wedge perto(robo, x, s).$$

Axiomas de estado sucessor (SSA) (D_{ss}) são axiomas que determinam o estado sucessor de um fluente na situação que sucede a execução de uma determinada ação. Cada fluente

relacional $F(\vec{x},s)$ que possui a forma

$$F(\vec{x},do(a,s)) \equiv \varphi_F(\vec{x},a,s),$$

em que $\varphi_F(\vec{x},a,s)$ são fórmulas de primeira ordem de variáveis livres que podem ter um efeito positivo ou negativo em relação ao fluente. Cada fluente relacional possui um SSA com a forma

$$a = PositiveAction \wedge \gamma^+(\vec{x},s) \vee \cdots \vee$$

$$F(\vec{x},s) \wedge \neg(a = NegativeAction \wedge \gamma^-(\vec{x},s) \vee \ldots),$$

onde $\gamma^+(\vec{x},s)$ representa o contexto em que o efeito do fluente pode ocorrer. São os responsáveis por determinar os valores verdadeiros de F na situação seguinte *do(a,s)* e por prover uma solução ao problema do quadro (*Frame Problem*) (REITER, 2001).

O problema do quadro consiste na dificuldade em se expressar o estado seguinte de um determinado fluente em função de cada ação que possa ser executada no domínio. No entanto, nem toda ação exerce influência sobre os valores de todo fluente. Ainda assim, é necessário que o axioma expresse a invariância do fluente frente à execução de tal ação. A solução descrita por Reiter (2001) consiste em negar todas as ações que exercem influência negativa sobre o valor de um determinado fluente, visto que o fluente seja verdadeiro na situação atual. Por exemplo, num domínio em que a única ação que exerça influência sobre a localização de um robô móvel seja a ação *move/3*, o SSA de um fluente *located/2* que define o local do robô em determinada situação seria definido pelo Axioma (82).

$$\begin{aligned} localizado(loc, do(a,s)) \leftrightarrow & (\exists t, loc_a)(a = move(loc_a, loc, t)) \wedge start(s,t') \wedge t' < t \\ & \vee localizado(loc, s) \wedge (\neg\exists t, loc_b)(a = move(loc, loc_b, t)). \end{aligned} \quad (82)$$

Axiomas de nome único (D_{una}) determinam que os axiomas do domínio são disjuntos par a par, ou seja, axiomas com o mesmo nome possuem os mesmos argumentos.

Base de dados inicial (D_{S_0}) é um conjunto de sentenças cujo único termo situacional é S_0 e determinam o início do problema, de onde todas as ações seguintes são subsequentes.

Axiomas fundamentais (Σ) são predicados funcionais do tipo *start(s,t)*, que representa o instante de tempo em que a situação s se inicia.

Considerando o domínio de um robô móvel, cuja única habilidade seja de se mover, pode-se axiomatizar o seu mundo conforme abaixo:

a) Ações primitivas:

$startMove(loc_a, loc_b, t)$, que representa a ação de início de movimento do local loc_a para o local loc_b no instante de tempo t.

$endMove(loc_a, loc_b, t)$, que representa a ação de término de movimento do local loc_a para o local loc_b no instante de tempo t.

b) Fluentes:

Relacional: $movendo(loc_a, loc_b, s)$, que representa que o robô está se movendo do local loc_a para o local loc_b na situação s.

Funcional: $localizado(loc_x, s)$, que representa que o robô está localizado no local loc_x na situação s.

c) Axiomas de precondição (D_{ap}):

A possibilidade de execução da ação $startMove(loc_a, loc_b, t)$ é verdadeira se e somente se o robô estiver localizado em loc_a na situação s, se ele já não estiver em movimento, e se a situação s se iniciar no instante de tempo t' que é anterior ao instante de tempo t que a ação ocorre. De forma similar, a possibilidade da ação $endMove(loc_a, loc_b, t)$ ser executada é verdadeira se e somente se o robô já estiver se movendo do local loc_a para o local loc_b na situação s, que se inicia no instante de tempo t', que é anterior ao instante de tempo t que a ação ocorre. Neste ponto poderiam existir outras verificações pertinentes à possibilidade de deslocamento do robô como por exemplo uma obstrução do caminho que será percorrido. No entanto, a intenção deste exemplo é ilustrar o arcabouço do Cálculo de Situações e não uma teoria completa de controle de um robô móvel.

$$
\begin{aligned}
Poss(startMove(loc_a, loc_b, t), s) &\leftrightarrow localizado(loc_a, s) \\
&\wedge (\neg\exists loc_c, loc_d) movendo(loc_c, loc_d, s) \\
&\wedge start(s,t') \wedge t' < t \\
Poss(endMove(loc_a, loc_b, t), s) &\leftrightarrow movendo(loc_a, loc_b, s) \\
&\wedge start(s,t') \wedge t' < t
\end{aligned}
$$

d) Axiomas de estado sucessor (D_{ss}):

O robô estará localizado no local loc_x se e somente se a ação executada for uma ação $endMove$ de forma que finalize um movimento do robô que partiu de um local loc_a para o local loc_x, ou que o robô já esteja localizado em loc_x e a ação executada não seja uma ação $startMove$ que inicie um movimento do robô do local loc_x para qualquer outro local. Similarmente, o robô estará se movendo de um local loc_a para um local loc_b se e somente se a ação executada seja uma ação $startMove$ que inicie o movimento desejado, ou que o robô já esteja realizando este movimento na situação s e a ação executada não seja uma ação $endMove$ que finalize este movimento.

$$
\begin{aligned}
&localizado(loc_x, do(a, s)) \leftrightarrow \\
&\qquad (\exists t, loc_a) a = endMove(loc_a, loc_x, t) \\
&\qquad \vee localizado(loc_x, s) \wedge (\neg\exists t, loc_b)(a = startMove(loc_x, loc_b, t)).
\end{aligned}
$$

$$
\begin{aligned}
&movendo(loc_a, loc_b, do(a, s)) \leftrightarrow \\
&\qquad (\exists t) a = startMove(loc_a, loc_b, t) \\
&\qquad \vee movendo(loc_a, loc_b, s) \wedge (\neg \exists t)(a = endMove(loc_a, loc_b, t)).
\end{aligned}
$$

e) Situação inicial (D_{S_0}):

A situação inicial s_0 se inicia no instante de tempo 0, e o robô está inicialmente localizado em loc_r.

$$
\begin{aligned}
&start(s_0, 0) \\
&localizado(loc_r, s_0)
\end{aligned}
$$

Uma vez que o domínio foi axiomatizado, pode-se fazer perguntas como:

A possibilidade de se movimentar do local loc_r para o local loc_f no instante de tempo 2 na situação s_0 é verdadeira? Valor esperado: Verdadeiro.

$$Poss(startMove(loc_r, loc_f, 2), s_0).$$

O robô estará se movimentando do local loc_r para o local loc_f na situação gerada após a execução da ação $startMove(loc_r, loc_f, 2)$? Valor esperado: Verdadeiro.

$$movendo(loc_r, loc_f, do(startMove(loc_r, loc_f, 2), s_0)).$$

O robô poderá finalizar o movimento do local loc_r para o local loc_f no instante de tempo 3 após ter iniciado movimento ? Valor esperado: Verdadeiro.

$$Poss(endMove(loc_r, loc_f, 3), do(startMove(loc_r, loc_f, 2), s_0)).$$

O robô ainda estará se movimentando após finalizar o movimento de loc_r para loc_f ? Valor esperado: Falso.

$$movendo(loc_r, loc_f, do(endMove(loc_r, loc_f, 3), do(startMove(loc_r, loc_f, 2), s_0))).$$

O cálculo de situações é utilizado como arcabouço para a escrita das relações espaciais qualitativas estudadas neste trabalho.

3.1.1 Raciocínio espacial qualitativo com relações topológicas

Em Bhatt, Rahayu e Sterling (2006) foi estudada a necessidade de representar raciocínio sobre ações e movimento para que se possa efetivamente raciocinar sobre as relações espaciais de um domínio. Por este motivo escolheram o RCC8 para ser codificado em ST (Cálculo de Situações) e efetivamente testar a teoria. O RCC8 foi a teoria escolhida por ser uma das teorias fundamentais de QSR e por sua simplicidade. A linguagem utilizada é formada por regiões (R) : r_1,r_2, . . . ,r_n, objetos (O) : o_1,o_2, . . . ,o_n e por situações (S) : s_1,s_2, . . . ,s_n. As relações

do RCC8 são representadas por $T_{rcc8} \equiv \{dc,ec,po,tpp,eq,tpp^{-1},ntpp,ntpp^{-1}\}$. Foram definidas formulações para o funcionamento do Cálculo de Situações como os axiomas de estado sucessor, axiomas de precondição, estado inicial do mundo e axiomas de nomes únicos.

O trabalho ilustrou o problema de ramificação ao definir consequências indiretas para as ações do mundo através restrições de estado de forma:

$$(\forall s).[dc(r_1,r_2,s) \wedge ec(r_2,r_3,s)] \supset [dr(r_1,r_3,s) \vee po(r_1,r_3,s) \vee pp(r_1,r_3,s)] \tag{83}$$

Por este motivo foi utilizado em Bhatt, Rahayu e Sterling (2006) uma função ternária *Caused/3* conforme Axioma (84).

$$\begin{aligned}(\forall s).[Holds(top(r_1,r_2),dc,s) \wedge Holds(top(r_2,r_3),ec,s)] \supset \\ Caused(top(r_1,r_3),\gamma,s),\end{aligned} \tag{84}$$

onde $\gamma \equiv [dr \vee po \vee pp]$.

A relação de causalidade foi utilizada para inferir relações espaciais combinando mudanças espaciais com aspectos causais. Desta forma, o axioma de estado sucessor foi formalizado conforme Fórmula (85).

$$\begin{aligned}&Poss(e,s) \wedge s' = Result(e,s) \supset \\ &[top(r_i,r_j,s') = t_{rcc8} \equiv (\exists o_i,o_j)\{e = tran(t_{rcc8},o_i,o_j) \wedge pos(o_i,s') = r_i \wedge pos(o_j,s') = r_j\} \\ &\quad \vee(\forall t'_{rcc8})(\exists o_i,o_j)\{pos(o_i,s) = r_i \wedge pos(o_j,s) = r_j \\ &\qquad \wedge top(r_i,r_j,s) = t_{rcc8} \wedge e \neq tran(t'_{rcc8},o_i,o_j)\}]\end{aligned} \tag{85}$$

A Fórmula (85) é lida como a possibilidade de ocorrência do evento e na situação s, e a possibilidade de existir uma situação s' resultante da ocorrência de e na situação s, é verdadeira quando existirem dois objetos o_i e o_j que ocupam o espaço delimitado pelas regiões r_i e r_j, realizarem o evento e de transição para uma das relações topológica definidas por t_{rcc8} na situação s', ou se já existir uma relação topológicas na situação s entre duas regiões r_i e r_j que representam espaços ocupados pelos objetos o_i e o_j e o evento e não for uma transição realizada pelos objetos para um estado não definido por t_{rcc8}.

$$Poss(tran(ec,o_i,o_j),s) \equiv [pos(o_i,s) = r_i \wedge pos(o_j,s) = r_j] \wedge [top(r_i, r_j, s) = dc \vee po]. \tag{86}$$

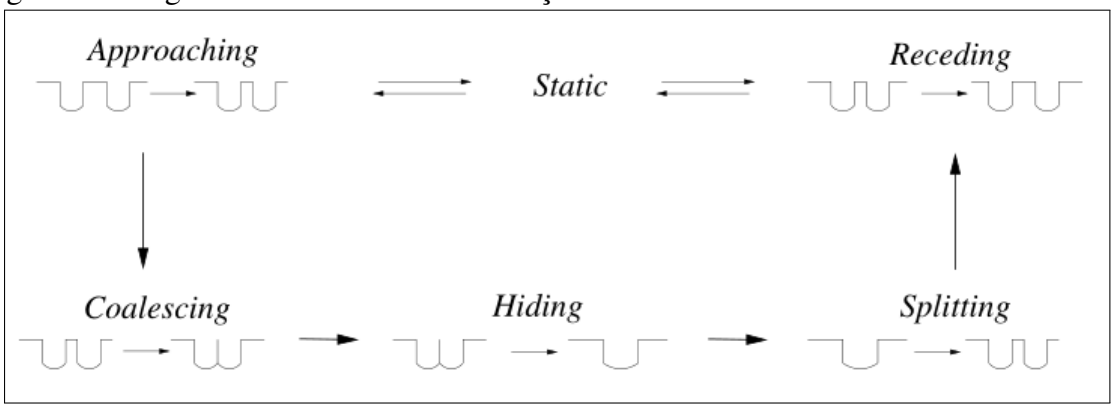
Assim, em conjunto com as outras definições sobre o domínio considerado no trabalho de Bhatt, Rahayu e Sterling (2006), informações iniciais do mundo e axioma de precondição a forma da Fórmula (86), ficou exemplificado um estudo de caso para representação e inferência de relações espaciais qualitativas utilizando o cálculo de situação como arcabouço, de forma similar como foi utilizado no trabalho de Soutchanski e Santos (2008), porém neste último com relações espaciais mais elaboradas.

3.2 RACIOCÍNIO SOBRE PERFIS DINÂMICOS DE PROFUNDIDADE

Em Soutchanski e Santos (2008) foi descrito um conjunto de axiomas para representação de conhecimento sobre objetos no espaço e seus movimentos do ponto de vista de um robô móvel. O presente trabalho apresenta uma implementação deste artigo escrito na linguagem de programação Prolog, que permite validar a teoria e contribuir para o seu entendimento. Uma implementação é importante pois muitas vezes ela revela detalhes não percebidos na definição teórica. Neste capítulo serão descritos teoricamente os axiomas apresentados em Soutchanski e Santos (2008) e os axiomas interpretados, os quais não foram formalmente descritos no artigo. A implementação será discutida no Capítulo 4.

A entidade básica do trabalho é o perfil de profundidade de campo de visão gerado através de um observador, conforme descrito na seção 2.1.4. Este perfil contém informações como disparidade de distância entre o objeto e o robô móvel, tamanho aparente, distância angular relativa e posicionamento em relação a outros objetos em cena, ou em relação ao próprio observador. Esta teoria utiliza as definições do cálculo de situações para formalização dos axiomas de transição em perfis de profundidade de campo, como também inclui os conceitos de geometria Euclidiana para descrever visibilidade e movimento dos objetos do mundo e do observador, que permanece sendo um único ponto de vista. Os picos de profundidade estudados no DPC Santos (2007) (conforme seção 2.1.4), foram estendidos em Soutchanski e Santos (2008) de forma que além da informação sensorial do robô, também são processadas informações sobre ações e movimento no mundo de forma que as transições do CND (Figura 9) passam a ser consequências lógicas da teoria.

Figura 9 – Diagrama conceitual de vizinhança - CND



Fonte: Soutchanski; Santos, 2008

A seguir será apresentada a formalização das relações espaciais da teoria $\mathcal{TDM}$.

3.2.1 Perfis de profundidade em cálculo de situações

O aspecto principal deste formalismo são as expressões definidas pelos axiomas de estado sucessor (SSA), teoricamente apresentados na seção 3.1, onde as relações que representam

as transições nas percepções de profundidade são combinadas com as ações que causaram tais transições. O objetivo é munir um robô móvel com a capacidade de descrever mudanças no mundo e raciocinar sobre os efeitos das ações tomadas por objetos no mundo e por suas próprias ações.

Além da representação de corpos (b) em cena, profundidade (u), tamanho (z) e distância angular (d), a linguagem compreende perfis de profundidade (p), pontos no tempo (t), ângulos de rotação (ω), direção (θ) em que o robô está posicionado em relação ao Norte, coordenadas (x) e (y) e pontos de vista (v) que correspondem a locais $loc(x,y)$. Os picos de profundidade são representados por $pk(b,u,z,d)$, interpretado como o pico do corpo b localizado em profundidade u do ponto de vista atual do robô móvel, possui tamanho z e distância angular d da borda esquerda do campo de visão.

A formalização da linguagem em cálculo de situações corresponde às ações primitivas que podem ser realizadas pelo robô móvel ou por objetos do domínio, fluentes relacionais para localização e movimento e transições para picos individuais ou entre pares de picos.

3.2.2 Ações primitivas

Ações primitivas são todas as ações possíveis no mundo. Pensando no domínio de um robô móvel, as ações possíveis são relacionadas a movimento ou percepção. Ações que compreendem movimento são $startMove(b,l_1,l_2,t)$, que representa que o corpo b está iniciando o movimento do local l_1 para o local l_2 no instante de tempo t, onde l_1 e l_2 são coordenadas x e y representadas por uma função $loc(x,y)$; $endMove(b,l_1,l_2,t)$, que representa o fim da movimentação de um corpo b do local l_1 para o local l_2 no instante de tempo t; $startPan(\omega,t)$, que representa que o robô iniciou uma rotação de ângulo ω no instante de tempo t; $endPan(\omega,t)$, que representa que o robô finalizou um rotação de ângulo ω no instante de tempo t. Valores de ω positivos significam que o robô realiza uma rotação no sentido horário. Valores de ω negativos significam que o robô realiza uma rotação no sentido anti-horário. A única ação relacionada à percepção que foi definida neste trabalho é $sense(p,loc(x_r,y_r),t)$, que representa que o robô localizado em $loc(x_r,y_r)$ efetuou uma leitura de seus sensores, gerando o perfil de profundidade p como resultado desta função no instante de tempo t. O perfil de profundidade p é composto de uma lista de picos $pk(b,u,z,d)$, onde cada pico representa um objeto localizado no seu campo de visão.

Cada ação possui uma precondição de execução. Não seria possível finalizar uma movimentação do local l_1 para l_3 ($endMove(Robot,l_1,l_3,time_2)$) se o robô estiver se movimentando de l_2 para l_4 ($startMove(Robot,l_2,l_4,time_1)$), tampouco seria possível finalizar uma movimentação num instante de tempo anterior ao seu início, que é o caso de $time_2 < time_1$. Para formalizar a precondição de execução das cinco ações primitivas, é necessário a definição de duas funções auxiliares básicas: a função $fieldView(\beta)$ que representa que o campo de visão do robô possui ângulo de abertura igual a β; e a função $visible(loc(x_r,y_r),b_1,\beta,\theta,s)$ que repre-

senta que o objeto b_1 está visível para o robô móvel que está localizado em $loc(x_r,y_r)$, que o ângulo de abertura do sensor do robô é β e que o robô está orientado θ graus em relação ao Norte na situação s. Com isso, pode-se formalizar as ações primitivas conforme Axiomas (87), (88), (89), (90) e (91).

$$Poss(startMove(b,l_1,l_2,t),s) \leftrightarrow location(b,l_1,s) \wedge \neg\exists l,l' moving(b,l,l',s) \wedge \\ l_1 \neq l_2 \wedge start(s,t') \wedge t \geq t'. \tag{87}$$

Da Fórmula (87) interpreta-se que a possibilidade do corpo b iniciar um movimento do local l_1 para o local l_2 no instante de tempo t na situação s é verdadeira, se e somente se o corpo b estiver localizado em l_1 em s, se não existir os locais l e l' de modo que o corpo b esteja se movendo de l para l' em s, se l_1 for diferente l_2, se o instante de tempo em que a situação s se inicia for t' e se t' for anterior ao instante t em que ação se desenvolve.

$$Poss(endMove(b,l_1,l_2,t),s) \leftrightarrow location(b,l_1,s) \wedge moving(b,l_1,l_2,s) \wedge l_1 \neq l_2 \wedge \\ start(s,t') \wedge t \geq t'. \tag{88}$$

A Fórmula (88) diz que a possibilidade de se finalizar o movimento do corpo b do local l_1 para l_2 é verdadeira se e somente se o corpo b estiver se movimentando de l_1 para l_2 na situação s. Para isso, o corpo b deve estar localizado em l_1 em s, l_1 e l_2 devem ser diferentes e o instante t em que a ação se desenvolve é posterior ao instante t' em que a situação s se inicia.

$$Poss(startPan(\omega,t),s) \leftrightarrow \neg\exists\omega' rotating(\omega',s) \wedge start(s,t') \wedge t \geq t'. \tag{89}$$

A Fórmula (89) representa que a possibilidade de iniciar uma rotação é verdadeira se o robô já não estiver em rotação e as restrições temporais e de situação forem atendidas.

$$Poss(endPan(\omega,t),s) \leftrightarrow rotating(\omega,s) \wedge start(s,t') \wedge t \geq t'. \tag{90}$$

A Fórmula (90) representa que a possibilidade de finalizar uma rotação é verdadeira se o robô já estiver em rotação e as restrições temporais e de situação forem atendidas.

$$Poss(sense(p,loc(x_r,y_r),t_2),s) \leftrightarrow start(s,t_1) \wedge t_2 \geq t_1 \wedge location(R,loc(x_r,y_r),s) \wedge \\ (\exists b,u,z,d,\theta,\beta) pk(b,u,z,d) \in p \wedge u > 0 \wedge z > 0 \wedge d > 0 \wedge facing(\theta,loc(x_r,y_r),s) \wedge \\ fieldView(\beta) \wedge visible(loc(x_r,y_r),b,\beta,\theta,s) \wedge \\ (\neg\exists b_I,u_I,z_I,d_I)(pk(b_I,u_I,z_I,d_I) \in p \wedge \neg visible(loc(x_r,y_r),b_I,\beta,\theta,s)). \tag{91}$$

A Fórmula (91) diz que a precondição para o robô efetuar uma ação sensorial é atendida se o perfil de profundidade p percebido possuir um pico de atributos positivos em respeito a

um objeto visível em seu campo de visão, e não possuir nenhum pico de objeto que não esteja atualmente visível dada a orientação do robô e o alcance de seu sensor. A formalização da função *visible/5* não foi explicitamente definida em Soutchanski e Santos (2008), no entanto, no presente trabalho foi implementada uma interpretação da função conforme Fórmula (92).

$$
\begin{aligned}
&visible(loc(x_r, y_r), b, \beta, \theta, do(a,s)) \leftrightarrow \\
&\quad (\exists t,p) a = sense(p,loc(x_r,y_r),t) \wedge pk(b,u,z,d) \in p \\
&\quad \vee (\exists t,x,y,x_1,y_1,L,e,\beta,\theta)(a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&\quad fieldView(\beta) \wedge facing(\theta, loc(x_1, y_1), s) \wedge location(b,loc(x_b,y_b),s) \wedge \\
&\quad location(R,loc(x_1,y_1),s) \wedge limitBoundary(L) \wedge \\
&\quad euD(loc(x,y),loc(x_r,y_r),e) \wedge e \leq L \wedge \\
&\quad angleV(loc(x_b, y_b),loc(x_r, y_r + 1), loc(x_r, y_r), \gamma) \wedge \\
&\quad abs(\theta - \gamma) \leq (\beta/2)) \\
&\quad \vee (\exists t,x_1,y_1,x_2,y_2,L,e)(a = endMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
&\quad fieldView(\beta) \wedge facing(\theta, loc(x_r, y_r), s) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad location(b,loc(x_1,y_1),s) \wedge limitBoundary(L) \wedge \\
&\quad euD(loc(x_r,y_r),loc(x_2,y_2),e) \wedge e \leq L \wedge \\
&\quad angleV(loc(x_2, y_2),loc(x_r, y_r + 1), loc(x_r, y_r), \gamma) \wedge \\
&\quad abs(\theta - \gamma) \leq (\beta/2)) \\
&\quad \vee visible(loc(x_r, y_r), b, \beta, \theta, s) \wedge location(b,loc(x_b,y_b),s) \wedge \\
&\quad (\neg\exists t,l,p',u',z',d',x_1,y_1)(a = endMove(R,loc(x_r,y_r),l,t) \\
&\quad\quad \vee a = endMove(b,loc(x_b,y_b),loc(x_1,y_1),t) \\
&\quad\quad \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
&\quad\quad\quad pk(b,u',z',d') \notin p').
\end{aligned}
\tag{92}
$$

Para que se possa verificar se um corpo é visível na situação inicial s_0, foi necessário criar uma formalização que recebesse a situação s_0 como parâmetro. Assim, é possível verificar se o axioma *visible/5* é verdadeiro após a execução da última ação executada na situação anterior, o que equivale à situação atual. Devido a situação inicial definida durante a implementação possuir um conjunto inicial de ações, o Axioma (93) não encontra um conjunto vazio de ações em s_0.

$$
\begin{aligned}
&visible(loc(x_r, y_r), b, \beta, \theta, s) \leftrightarrow \\
&\quad (\exists ls,la) ls = lastSituation(s) \wedge la = lastAction(s) \wedge \\
&\quad visible(loc(x_r, y_r), b, \beta, \theta, do(la, ls)).
\end{aligned}
\tag{93}
$$

É interessante notar que na Fórmula (92) não há uma formalização sobre as ações de rotação. Isto ocorre devido ao axioma receber os ângulos β e θ como parâmetro, que representam respectivamente a abertura do campo de visão dos sensores do robô e a direção em que o robô se encontra em relação ao Norte. Desta forma, mesmo que exista um movimento ou uma rotação torna-se possível verificar se um corpo é visível ou não do ponto de vista do robô através da utilização dos valores atuais de cada parâmetro.

3.2.3 Movimento

As ações primitivas governam o conhecimento sobre as relações físicas no domínio do robô móvel. Através delas, é possível inferir relações em respeito a ações e mudanças no mundo. Os Axiomas *facing/3*, *location/3*, *moving/4* e *rotating/2* pertencem a esta categoria. Informalmente, o fluente $facing(\theta,loc(x_r,y_r),s)$ representa que o robô que está localizado em $loc(x_r,y_r)$ e possui orientação de ângulo θ em relação ao Norte na situação s. O fluente $location(b,loc(x_b,y_b),s)$ representa que o corpo b está localizado no local $loc(x_b,y_b)$ em s. O fluente $moving(b,l_1,l_2,s)$ representa que o corpo b está se movendo do local l_1 para o local l_2 na situação s. O fluente $rotating(\omega,s)$, representa que o robô móvel está realizando uma rotação de ângulo ω na situação s. A formalização dos axiomas destes fluentes pode ser observada nos axiomas de estado sucessor (SSA) (94) a (97).

$$\begin{aligned} facing(\theta_2,loc(x_r,y_r),do(a,s)) \leftrightarrow (\exists t,\omega,\theta_1)(a = endPan(\omega,t) \wedge \\ facing(\theta_1,loc(x_r,y_r),s) \wedge \theta_2 = \theta_1 + \omega) \vee facing(\theta_2,loc(x_r,y_r),s) \wedge \\ (\neg\exists t,\omega)(a = endPan(\omega,t)). \end{aligned} \tag{94}$$

Formalmente, para que o robô esteja direcionado a um ângulo θ_2 em relação ao Norte após a execução de uma ação a na situação s do mundo, é preciso que o robô esteja localizado em $loc(x_r,y_r)$ em s, e que esta ação a seja uma ação *endPan/2* na qual a soma do ângulo ω de rotação com o ângulo θ_1 em que o robô estava direcionado anteriormente é igual a θ_2, ou, que o robô já esteja orientado em θ_2 graus e a ação a não seja nenhuma ação que tenha relação direta com sua orientação, no caso, a é diferente de *endPan/2*. A teoria considera que todos os passos de rotação e movimentação são curtos o suficiente para que a percepção de mudança de posição ou direção ocorra para efeitos de cálculo apenas nas ações de término de movimento e rotação. Por este motivo caso a ação executada seja *startPan/2* ou *startMove/4*, o axioma *facing/3* persiste para o último ângulo θ conhecido.

$$\begin{aligned} location(b,loc(x,y),do(a,s)) \leftrightarrow (\exists t,x',y')a = endMove(b,loc(x',y'),loc(x,y),t) \vee \\ location(b,loc(x,y),s) \wedge (\neg\exists t,x_2,y_2)(a = endMove(b,loc(x,y),loc(x_2,y_2),t)). \end{aligned} \tag{95}$$

Interpreta-se de (95) que a localização de um robô é $loc(x,y)$ após uma ação ser efetuada se esta ação for um término de movimento *endMove/4* de algum outro lugar $loc(x',y')$ para

$loc(x,y)$, ou se o robô já estiver nesta posição e a ação não indique que o robô tenha se movido para um lugar $loc(x_2,y_2)$ diferente. Na situação inicial da implementação foram definidos valores iniciais para o robô móvel e para os corpos do domínio, devido não ter sido desenvolvido durante este projeto um sistema de visão, que pudesse gerar os perfis de profundidade necessários para que o robô tivesse a percepção destes corpos.

$$moving(b,l_1,l_2,do(a,s)) \leftrightarrow (\exists t)a = startMove(b,l_1,l_2,t) \vee moving(b,l_1,l_2,s) \wedge \\ (\neg\exists t)(a = endMove(b,l_1,l_2,t)). \tag{96}$$

O Axioma (96) diz que o corpo b estará se movendo caso realize uma ação de início de movimento *startMove/4* ou no caso de ele já estar em movimento e a ação não o finalize.

$$rotating(\omega,do(a,s)) \leftrightarrow (\exists t)a = startPan(\omega,t) \vee rotating(\omega,s) \wedge \\ (\neg\exists t)(a = endPan(\omega,t)). \tag{97}$$

Analogamente, o robô estará em rotação após uma ação a caso esta seja uma ação de início de rotação ou se ele já estiver em rotação e a ação não finalize o movimento.

3.2.4 Atributos

Ações não geram apenas mudanças topológicas na localização e orientação dos objetos no domínio, quando levamos em conta a percepção do mundo que um robô móvel tem de seu ponto de vista egocêntrico, também ocorrem alterações na percepção de *tamanho*, *disparidade* e *distância* entre os objetos. Os fluentes *depth/4*, *size/4* e *dist/5* são responsáveis por representar tais atributos dos objetos observados. De forma auxiliar, para a formalização destes axiomas, foi utilizada uma função *euD/3* para determinar a distância Euclidiana entre dois pontos.

O predicado $depth(pk(b,u,z,d),u,loc(x_r,y_r),do(a,s))$, caso verdadeiro, representa que a profundidade de um pico vale u após a execução de uma ação a, se e somente se a for uma ação sensorial que gerou um perfil de profundidade que contenha um pico do objeto b de profundidade u, ou se a for uma ação de término de movimento de tal forma que a distância Euclidiana entre a localização do objeto e do robô resultem em u. Formalmente, o axioma de estado sucessor para este fluente é a Fórmula (98). Lembrando que a teoria considera que os passos de movimento e rotação são curtos o suficiente para que a alteração de valores para fins de cálculo ocorra no término do movimento. Desta forma, enquanto houver movimento o valor resultante de profundidade deste Axioma permanece inalterado.

$$depth(pk(b,u,z,d),u,loc(x_r,y_r),do(a,s)) \leftrightarrow \\ (\exists t,p)a = sense(p,loc(x_r,y_r),t) \wedge pk(b,u,z,d) \in p$$

$$
\begin{aligned}
&\vee\ (\exists t,x,y,x_1,y_1,r,e)(a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&\quad location(b,loc(x,y),s) \wedge location(R,loc(x_1,y_1),s) \wedge radius(b,r) \wedge \\
&\quad euD(loc(x,y),loc(x_r,y_r),e) \wedge (u = e - r)) \\
&\vee\ \exists t,x_1,y_1,x_2,y_2,r,e)(a = endMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge radius(b,r) \wedge \\
&\quad euD(loc(x_r,y_r),loc(x_2,y_2),e) \wedge (u = e - r)) \\
&\vee\ depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad (\exists x,y)location(b,loc(x,y),s) \wedge \\
&\quad (\neg\exists t,l,p',u',z',d',x_1,y_1)(a = endMove(R,loc(x_r,y_r),l,t) \\
&\qquad \vee\ a = endMove(b,loc(x,y),loc(x_1,y_1),t) \\
&\qquad \vee\ a = sense(p',loc(x_r,y_r),t) \wedge \\
&\qquad\quad pk(b,u',z',d') \in p' \wedge u \neq u').
\end{aligned}
\tag{98}
$$

Em Soutchanski e Santos (2008) não foram explicitamente definidos os axiomas de estado sucessor para os fluentes *size/4* e *dist/5*, consta apenas a informação de que seriam análogos ao SSA de *depth/4*. Assim, podemos considerar de forma similar, que o SSA para *size/4* seja formalmente definido pelo Axioma (99).

$$
\begin{aligned}
&size(pk(b,u,z,d),z,loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists t,p)a = sense(p,loc(x_r,y_r),t) \wedge pk(b,u,z,d) \in p \\
&\vee\ (\exists t,x,y,x_1,y_1,r,e)(a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&\quad location(b,loc(x,y),s) \wedge location(R,loc(x_1,y_1),s) \wedge \\
&\quad radius(b,r) \wedge euD(loc(x,y),loc(x_r,y_r),e) \wedge (z = 2 \cdot \arcsin(r/e) \cdot 180/\pi)) \\
&\vee\ (\exists t,x_1,y_1,x_2,y_2,r,e)(a = endMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge \\
&\quad radius(b,r) \wedge euD(loc(x_r,y_r),loc(x_2,y_2),e) \wedge (z = 2 \cdot \arcsin(r/e) \cdot 180/\pi)) \\
&\vee\ size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad (\exists x,y)location(b,loc(x,y),s) \wedge \\
&\quad (\neg\exists t,l,p',b',u',z',d',x_1,y_1)(a = endMove(R,loc(x_r,y_r),l,t) \\
&\qquad \vee\ a = endMove(b,loc(x,y),loc(x_1,y_1),t) \\
&\qquad \vee\ a = sense(p',loc(x_r,y_r),t) \wedge pk(b',u',z',d') \in p' \wedge z \neq z').
\end{aligned}
\tag{99}
$$

De maneira geral, os axiomas de estado sucessor definem que a situação seguinte a uma situação é calculada por uma ação direta que tenha influência sobre o atributo ou que seu valor

já seja de conhecimento pela situação anterior e que a ação não tenha efeito negativo sobre o fluente.

O terceiro fluente de atributo é o *dist/5*. Semelhante ao fluente que determina profundidade *depth/4*, também podemos definir o axioma de estado sucessor para *dist/5*.

A formalização pode ser encontrada na Fórmula (100).

$$
\begin{aligned}
&dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists t,p) a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge \\
&\quad pk(b_b,u_b,z_b,d_b) \in p \wedge ((d_b - z_b > d_a \wedge d = d_b - d_a - z_b) \\
&\quad\quad \vee (d_a - z_a > d_b \wedge d = d_a - d_b - z_a)) \\
&\vee (\exists t,x_a,y_a,x_b,y_b,x_1,y_1,r,e)(a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
&\quad location(R,loc(x_1,y_1),s) \wedge size(pk(b_a, u_a, z_a, d_a), z_a, loc(x_r, y_r), s) \wedge \\
&\quad size(pk(b_b, u_b, z_b, d_b), z_b, loc(x_r, y_r), s) \wedge \\
&\quad angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r), \gamma) \wedge (d = \gamma - (z_a/2) - (z_b/2))) \\
&\vee (\exists t,x_a,y_a,x_b,y_b,x_1,y_1,r,e)(a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge \\
&\quad location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge size(pk(b_a, u_a, z_a, d_a), z_a, loc(x_r, y_r), s) \wedge \\
&\quad size(pk(b_b, u_b, z_b, d_b), z_b, loc(x_r, y_r), s) \wedge \\
&\quad angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r), \gamma) \wedge (d = \gamma - (z_a/2) - (z_b/2))) \\
&\vee (\exists t,x_a,y_a,x_b,y_b,x_1,y_1,r,e)(a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge size(pk(b_a, u_a, z_a, d_a), z_a, loc(x_r, y_r), s) \wedge \\
&\quad size(pk(b_b, b_b, z_b, d_b), z_b, loc(x_r, y_r), s) \wedge \\
&\quad angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r), \gamma) \wedge (d = \gamma - (z_a/2) - (z_b/2))) \\
&\vee dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge (\neg\exists t,l_{r2},l_{a2},l_{b2},p',b',u',z',d') \\
&\quad (a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(a,loc(x_a,y_a),l_{a2},t) \\
&\quad \vee a = endMove(b,loc(x_b,y_b),l_{b2},t) \\
&\quad \vee a = sense(p',loc(x_r,y_r),t) \wedge pk(b'_a,u'_a,z'_a,d'_a) \in p') \wedge \\
&\quad\quad pk(b'_b,u'_b,z'_b,d'_b) \in p' \wedge \\
&\quad\quad\quad ((d'_b - z'_b > d'_a \wedge d' = d'_b - d'_a - z'_b \wedge d \neq d') \\
&\quad\quad\quad \vee (d'_a - z'_a > d'_b \wedge d' = d'_a - d'_b - z'_a \wedge d \neq d'))).
\end{aligned}
\tag{100}
$$

Para a definição do predicado *dist/5* foi necessária a utilização de uma função chamada *angleV/4* que calcula o ângulo aparente entre dois picos. Nesta função, o cálculo foi realizado através de vetores gerados da localização do robô em direção a cada um dos picos. A sua formalização pode ser observada no Axioma (101).

$$
\begin{aligned}
&angleV(loc(X_a, Y_a), loc(X_b, Y_b), loc(X_r, Y_r), \gamma) \leftrightarrow \\
&\qquad V_{ax} = X_a - X_r \wedge V_{ay} = Y_a - Y_r \wedge \\
&\qquad V_{bx} = X_b - X_r \wedge V_{by} = Y_b - Y_r \wedge \\
&\qquad \gamma_t = \arccos\left(\frac{(V_{ax} \cdot V_{bx}) + (V_{ay} \cdot V_{by})}{\sqrt{V_{ax}^2 + V_{ay}^2} \cdot \sqrt{V_{bx}^2 + V_{by}^2}}\right) \cdot \frac{180}{\pi} \wedge \\
&\qquad \gamma = abs(\gamma_t).
\end{aligned}
\tag{101}
$$

O artigo de Soutchanski e Santos (2008) menciona que o predicado *dist/5* calcula a distância angular entre dois picos de profundidade ou entre um pico e a borda esquerda do campo de visão do observador. No entanto, não foram explicitamente definidos os detalhes do axioma deste predicado. Assim, em complemento ao predicado *dist/5* que foi descrito na Fórmula (100), foi formulado um predicado *dist/4* para simplificar o caso em que se efetua o cálculo de distância angular de um pico e a borda lateral esquerda do campo de visão. Segue Fórmula (102) que formaliza tal funcionalidade.

$$
\begin{aligned}
&dist(pk(b,u,z,d),d,loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\qquad (\exists p,t) a = sense(p,loc(x_r,y_r),t) \wedge pk(b,u,z,d) \in p \\
&\quad \vee (\exists x,y,x_1,y_1,t,r,\beta,\gamma,\theta) a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&\qquad location(R,loc(x_1,y_1),s) \wedge location(b,loc(x,y),s) \wedge \\
&\qquad size(pk(b, u_b, z_b, d_b), z_b, loc(x_r, y_r), s) \wedge \\
&\qquad fieldView(\beta) \wedge facing(\theta,loc(x_r,y_r),s) \wedge \\
&\qquad angleV(loc(x,y),loc(x_r, y_r + 1),loc(x_r,y_r),\gamma) \wedge \\
&\qquad d = (\beta/2) + \gamma - \theta - (z_b/2) \\
&\quad \vee (\exists x_1,y_1,x_2,y_2,t,r,\beta,\gamma,\theta) a = endMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
&\qquad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge \\
&\qquad size(pk(b, u_a, z_a, d_a), z_a, loc(x_r, y_r), s) \wedge \\
&\qquad fieldView(\beta) \wedge facing(\theta,loc(x_r,y_r),s) \wedge \\
&\qquad angleV(loc(x_2,y_2),loc(x_r, y_r + 1),loc(x_r,y_r),\gamma) \wedge \\
&\qquad d = (\beta/2) + \gamma - \theta - (z_a/2)
\end{aligned}
\tag{102}
$$

$$\vee\ (\neg\exists p',u',z',d',t,l_{r2},l_{b2})(a = endMove(R,loc(x_r,y_r),l_{r2},t)$$
$$\vee\ a = endMove(b,loc(x,y),l_{b2},t)$$
$$\vee\ a = sense(p',loc(x_r,y_r),t) \wedge pk(b,u',z',d') \in p' \wedge d \neq d' \wedge$$
$$location(R,loc(x_r,y_r),s) \wedge location(b,loc(x,y),s) \wedge$$
$$dist(pk(b,u,z,d),d,loc(x_r,y_r),s).$$

3.2.5 Transições de pico individual

Uma vez que os atributos inerentes aos picos de profundidade observados foram definidos, pode-se representar transições entre estados. Tais transições podem ser definidas para picos individuais ou entre pares de picos.

Um pico de profundidade pode crescer (*extending/3*), diminuir (*shrinking/3*), aparecer (*appearing/3*), desaparecer (*vanishing/3*) ou se manter estático (*peak_static/3*). Os axiomas de estado sucessor dessas transições expressam a percepção do ponto de vista de um robô móvel sobre as relações espaciais dos objetos em seu campo de visão devido às suas ações e devido às ações dos objetos. A formalização dos axiomas SSA para transições de pico individual são descritos conforme Fórmula (103) neste capítulo, e Fórmulas (107), (108), (109) e (110) no Apêndice A que por motivos de extensão foram omitidos aqui.

$$\begin{aligned}
&extending(pk(b,u,z,d),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists p,t,u',z',d')a = sense(p,loc(x_r,y_r),t) \wedge \\
&\quad pk(b,u',z',d') \in p \wedge location(R,loc(x_r,y_r),s) \wedge limitBoundary(L) \wedge \\
&\quad size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge \\
&\quad z < z' \wedge d \leq L \wedge d' \leq L \\
&\vee\ (\exists x,y,x_1,y_1,t,r,e,L,u',z')a = startMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&\quad location(R,loc(x_1,y_1),s) \wedge location(b,loc(x,y),s) \wedge limitBoundary(L) \wedge \\
&\quad radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_1,y_1),s) \wedge \\
&\quad depth(pk(b,u,z,d),u,loc(x_1,y_1),s) \wedge euD(loc(x,y),loc(x_r,y_r),e) \wedge \\
&\quad z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z < z' \wedge d \leq L \wedge d' \leq L \\
&\vee\ (\exists x_1,y_1,x_2,y_2,t,r,e,L,u',z')a = startMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge limitBoundary(L) \wedge \\
&\quad radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge \\
&\quad depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge euD(loc(x_2,y_2),loc(x_r,y_r),e) \wedge \\
&\quad z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z < z' \wedge d \leq L \wedge d' \leq L
\end{aligned} \tag{103}$$

$$\vee\ (\neg\exists t,l_{r2},l_{b2},p',u',z',d')(a = endMove(R,loc(x_r,y_r),l_{r2},t)$$
$$\vee\ a = endMove(b,loc(x,y),l_{b2},t)$$
$$\vee\ a = sense(p',loc(x_r,y_r),t) \wedge pk(b,u',z',d') \in p' \wedge (z \geq z' \vee d' > L)) \wedge$$
$$location(R,loc(x_r,y_r),s) \wedge location(b,loc(x,y),s) \wedge$$
$$extending(pk(b,u,z,d),loc(x_r,y_r),s).$$

O Axioma (103) $extending(pk(b,u,z,d),loc(x_r,y_r),s)$ representa que um pico de profundidade estará crescendo se existir uma ação de sensoriamento realizada de tal forma que o pico presente no perfil de profundidade gerado é maior do que numa situação anterior, ou se existir uma ação de movimentação do robô ou do objeto representado pelo pico de profundidade de modo que a nova localização dos objetos permite calcular um tamanho angular do pico maior do que na situação que precede a ação. É preciso notar que em ambas as situações analisadas é necessário que a profundidade u em s e u' em $do(a,s)$ sejam menores do que o limiar L (constante do sistema que precisa ser definida durante testes reais no domínio) que define o alcance máximo dos sensores do robô R. A relação de profundidade de um pico com o limiar L é o que diferencia um pico que está crescendo de um pico que está aparecendo. Assim, caso a profundidade fosse maior ou igual a L em s, $u \geq L$, e após determinada ação a profundidade torna-se menor do que L, $u' < L$, então o pico seria percebido como *appearing/3* e não como *extending/3*.

Para os Axiomas *appearing/3* e *vanishing/3* existe a necessidade de um segundo limiar de profundidade que será chamado de *thresholdBoundary(TB)*. Este limiar remete à Santos (2007), apesar de não ter sido mencionado em (SOUTCHANSKI; SANTOS, 2008). Sua função é determinar uma pequena faixa de valor entre o alcance máximo dos sensores do robô e o ponto de separação entre *appearing/extending* e entre *shrinking/vanishing*.

De forma análoga, o Axioma (108) $shrinking(pk(b,u,z,d),loc(x_r,y_r),s)$ determina que um pico está diminuindo se existir uma ação de sensoriamento de tal forma que o pico presente no perfil de profundidade gerado é menor do que na situação anterior, ou se existir uma ação de movimentação do robô de modo que a nova localização dos objetos permita calcular um tamanho angular menor do pico do que era na situação antecessora. Em ambas as situações é necessário que a profundidade do pico seja menor do que L.

Caso a profundidade calculada ou notada seja maior que L após a execução de uma ação, o pico será interpretado como *vanishing/3* (Axioma (109)) e não como *shrinking/3* (Axioma (108)).

Por fim, caso a noção de profundidade do pico permaneça inalterada de uma situação para outra, o axioma que descreve tal situação é *peak_static/3* (Axioma (110)).

3.2.6 Transições entre pares de picos

Irão existir em determinados momentos múltiplos picos de profundidade no campo de visão do observador. A teoria proposta em Soutchanski e Santos (2008) também define relações espaciais entre picos. As relações previstas nesta categoria ditam que dois picos de profundidade podem se aproximar (*approaching/4*), se afastar (*receding/4*), se fundir (*coalescing/4*), se ocluir (*hiding/4*), se separar (*splitting/4*) ou se manterem estáticos entre si (*two_peak_static/4*). Por motivos de extensão, os demais axiomas foram omitidos, porém estão descritos no Apêndice A.

$$
\begin{aligned}
&approaching(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists p,t,\Delta,d')a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge \\
&\quad pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\quad ((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge \\
&\quad thresholdUnion(\delta) \wedge d' < d \wedge d' \geq \Delta \\
&\vee (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,e_b,z'_a,z'_b,\gamma) \\
&\quad a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
&\quad radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge \\
&\quad euD(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge euD(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge \\
&\quad z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
&\quad angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
&\quad d' = \gamma - (z'_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
&\vee (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,z'_a,\gamma) \\
&\quad a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
&\quad radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\quad euD(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \qquad (104) \\
&\quad angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
&\quad d' = \gamma - (z'_a/2) - (z_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
&\vee (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_b,z'_b,\gamma) \\
&\quad a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge \\
&\quad radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge
\end{aligned}
$$

$$
\begin{aligned}
& euD(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
\vee\ & (\neg\exists p',t,\Delta,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b) \\
& (a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(b_a,loc(x_a,y_a),l_{a2},t) \\
& \quad \vee a = endMove(b_b,loc(x_b,y_b),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
& \quad pk(b_a,u'_a,z'_a,d'_a) \in p' \wedge pk(b_b,u'_b,z'_b,d'_b) \in p' \wedge \\
& \quad ((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge \\
& \quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& \quad thresholdUnion(\Delta) \wedge (d' \geq d \vee d' < \Delta)) \wedge \\
& location(R,loc(x_r,y_r),s) \wedge location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
& approaching(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s).
\end{aligned}
$$

O Axioma (104) $approaching(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s)$ determina que dois picos estão se aproximando caso exista uma ação de sensoriamento de tal forma que os picos presentes no perfil de profundidade gerado estão a uma distância angular um do outro menor do que estavam na situação anterior, ou se existir uma ação de movimentação do robô ou de qualquer um dos objetos a e b de tal forma que a nova localização dos objetos permite calcular uma distância angular entre os picos que seja menor na nova situação do que era antes da execução da ação. É necessário notar também que a distância angular calculada ou percebida deve ser maior do que um limiar Δ que faz a diferenciação das relações *approaching/4* e *coalescing/4*.

De forma análoga, o Axioma (114) $receding(pk(b_a,u_a,z_a,d_a),p(b_b,u_b,z_b,d_b),loc(x_r.y_r),s)$ determina que dois picos estão se afastando se existir uma ação de sensoriamento de tal forma que os picos presentes no perfil de profundidade gerado estão a uma distância angular um ao outro maior do que estavam antes da execução da ação, ou se existir uma ação de movimentação do robô ou de qualquer um dos dois objetos de modo que a nova localização dos objetos permite calcular uma distância angular entre os picos que seja maior na nova situação do que na situação antecessora. Tal distância precisa ser maior que o limiar Δ .

Caso contrário, a interpretação que se tem quando a distância angular na nova situação é menor que o limiar Δ , é de que os picos estarão se separando (*splitting/4*) (Axioma 113) e não apenas se afastando (*receding/4*) (Axioma 114).

Similarmente, foi definido em Soutchanski e Santos (2008) que um terceiro limiar define a diferença de relações *coalescing/4* para *hiding/4*. A interpretação direta que se tem do artigo é de que a relação *hiding/4* é verdadeira se existir uma sobreposição quase completa de corpos. Desta forma, o Axioma (105) formaliza tal relação.

$$
hiding(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow
$$

$$
\begin{aligned}
&(\exists p,t,\Delta,d',x_a,y_a,x_b,y_b,e_a,e_b) a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge \\
&pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge \\
&dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge \\
&d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
&location(b_a, loc(x_a,y_a), s) \wedge location(b_b, loc(x_b,y_b), s) \wedge \\
&euD(loc(x_a, y_a), loc(x_r, y_r), e_a) \wedge euD(loc(x_b, y_b), loc(x_r, y_r), e_b) \wedge \\
&(e_a >= e_b \wedge d > z_a \cdot (-1) \vee e_b > e_a \wedge d > z_b \cdot (-1)) \\
\vee\ &(\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,e_b,z'_a,z'_b,\gamma) \\
&a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge \\
&location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
&radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge \\
&euD(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge euD(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge \\
&z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
&angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
&d' = \gamma - (z'_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
&(e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1)) \\
\vee\ &(\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,z'_a,\gamma) \\
&a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
&radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&euD(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \\
&angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
&d' = \gamma - (z'_a/2) - (z_b/2) \wedge d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
&(e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z_b \cdot (-1)) \\
\vee\ &(\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_b,z'_b,\gamma) \\
&a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge \\
&radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&euD(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
&angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
&d' = \gamma - (z_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
&(e_a >= e_b \wedge d_a > z_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1)) \\
\vee\ &(\neg\exists p',t,\Delta,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b)
\end{aligned}
\tag{105}
$$

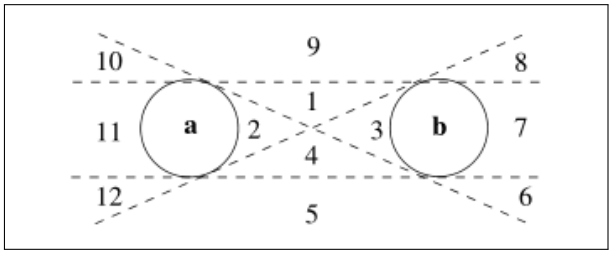
$$
\begin{aligned}
&(a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(b_a,loc(x_a,y_a),l_{a2},t) \\
&\quad \vee a = endMove(b_b,loc(x_b,y_b),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
&\qquad pk(b_a,u'_a,z'_a,d'_a) \in p' \wedge pk(b_b,u'_b,z'_b,d'_b) \in p' \wedge \\
&\qquad ((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge \\
&\qquad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\qquad thresholdUnion(\Delta) \wedge (d' \geq d \vee d' \geq \Delta)) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
&\quad hiding(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s).
\end{aligned}
$$

Por fim, caso as relações entre os picos se mantenham ao longo do tempo, a percepção que se tem conforme a teoria é de que os mesmos estão estáticos (*two_peak_static/4*) (Axioma 116).

3.2.7 Percepção de movimento

Nas seções anteriores foram definidos os axiomas de estado sucessor das transições para picos em perfis de profundidade. O conjunto de todos os axiomas de fluentes, representados por D_{SS}, somados aos axiomas de precondição D_{ap}, uma base de dados inicial D_{S_0} e com D_{una} e Σ formam a teoria chamada Teoria de Profundidade e Movimento ($\mathcal{TDM}$) (SOUTCHANSKI; SANTOS, 2008).

Figura 10 – Bi tangentes entre dois objetos



Fonte: Soutchanski; Santos, 2008

Levando-se em consideração um domínio onde o observador e objetos podem se mover, as percepções descritas pela teoria são *corretas* e *completas* em respeito a movimento (SOUTCHANSKI; SANTOS, 2008). A prova deste teorema dita que para cada ação do mundo que gere uma transição, se um fluente F passa a ser falso então F' passa a ser verdadeiro. Além disso, para cada movimento realizado no domínio existe uma ação a mencionada nos SSAs que corresponde a este movimento. Essa afirmação se justifica com base na Figura 10 que ilustra todas as doze regiões que determinam os pontos de vista distintos qualitativamente possíveis.

No Capítulo 4 serão descritas as notas de implementação destes axiomas e um experimento simulado onde um robô móvel passa mais da metade das regiões definidas pelas bi tangentes, pois as regiões que não foram exploradas são análogas.

4 IMPLEMENTAÇÃO

Neste capítulo serão descritos os detalhes referentes à implementação da teoria $\mathcal{TDM}$ Soutchanski e Santos (2008) discutida no Capítulo 3. Os axiomas descritos no capítulo anterior foram implementados em linguagem de programação *Prolog* por permitir uma tradução quase direta. Alguns detalhes dos axiomas foram modificados durante a implementação e serão detalhados a seguir.

O cálculo de situações define que cada ação realizada no mundo leva a uma nova situação através do predicado *do(a,s)* (REITER, 2001). Desta forma, uma situação é uma sequência de ações da forma *do(c,do(b,do(a,s))))*, que representa que as ações a, b e c foram executadas uma após a outra, e que não existe nenhuma outra ação realizada entre elas. Teoricamente, quando consideramos algumas poucas ações no mundo, é fácil enxergar a sequência de ações e analisar os acontecimentos no mundo. No entanto, pensando no domínio de um robô móvel, as ações se acumulam rapidamente e, como o processamento remete sempre ao estado inicial, ele se torna cada vez mais condicionalmente custoso ao longo do tempo. Assim, foi efetuada uma tentativa de adotar uma estrutura que atualiza a situação corrente após cada ação, visando amenizar esta questão. As ações executadas são armazenadas em uma lista secundária para consulta e raciocínio sobre o passado.

Quando a teoria fala de movimento de um local $loc(x_a, y_a)$ para uma local $loc(x_b, y_b)$, a interpretação que se teve é de que são posições relativas apenas ao robô móvel. O sistema sensorial do robô, que espera-se que seja um sistema de câmera estéreo ou laser, obtém os dados de um perfil de profundidade do campo de visão do robô. Com isso, e tendo conhecimento sobre os dados técnicos dos sensores sobre a correspondência em distância real em uma determinada unidade de medida que os dados do sinal obtido representam, assume-se que é possível determinar que um objeto está localizado a y unidades de distância do robô e deslocado x unidades de distância à direita ou esquerda.

O código foi desenvolvido utilizando o *Prolog Development Tool for Eclipse* na versão 3.1, que é uma extensão da interface de desenvolvimento *Eclipse* para se trabalhar com o Prolog, e o interpretador *SWI-Prolog* na versão 7.2.3. Os axiomas foram separados em módulos e organizados pelos tipos fundamental, auxiliar, de movimento, de atributo, de transição para pico individual e de transição para pares de pico. Nas próximas duas seções serão descritos detalhes sobre a implementação com e sem atualização da situação corrente em memória.

4.1 MÉTODO COM ATUALIZAÇÃO DE SITUAÇÃO EM MEMÓRIA

Este método é uma alternativa ao processamento tradicional do Cálculo de Situações e foi desenvolvido com o intuito de separar o conhecimento da situação corrente em duas bases de dados. Uma base de dados contém informação de ações recentes que ocorreram no domínio e que será usada para o raciocínio sobre ações na situação corrente. A segunda base de da-

dos contém informação de ações passadas que não exercem mais interferência sobre o cálculo dos axiomas de estado sucessor devido ações mais recentes, mas que devem permanecer em memória para consulta de ações passadas.

4.1.1 Axiomas fundamentais

Como parte dos axiomas fundamentais foi utilizada a fórmula:

$$start(do(A,S),T) : -time(A,T).$$

que representa que o instante de tempo t em que a ação A se inicia determina o instante de tempo t que a situação $do(A,S)$ se inicia. Adicionalmente, o instante de tempo em que uma ação primitiva se inicia é representado por:

$$\begin{gathered} time(Action,Time) : - \\ Action = ..List, \\ last(List,Time). \end{gathered}$$

Uma das funções básicas elaboradas é a *situation/2* que representa a lista de ações executadas em determinada situação. Neste trabalho, a situação inicial foi nomeada $s0$ e sempre que se faz referência à situação corrente, estará sendo feita referência à situação $s0$. Em Prolog, a implementação desta função pode ser expressa por

$$situation(S, A).,$$

que representa a lista de ações A da situação S.

O predicado principal é *do(a,s)*. Este é a responsável por gerenciar as ações que ocorrem no domínio visto que sua função é gerar uma nova situação, após a execução de uma ação a na situação s. No Quadro 1 pode-se observar que a função verifica se o axioma de precondição da ação é verdadeiro antes da ação ser propriamente executada. Em seguida, caso a ação possa ser executada, a ação é adicionada à lista de ações da situação atual, as consequências lógicas da ação são atualizadas e o instante de tempo em que a situação atual se encontra é modificado.

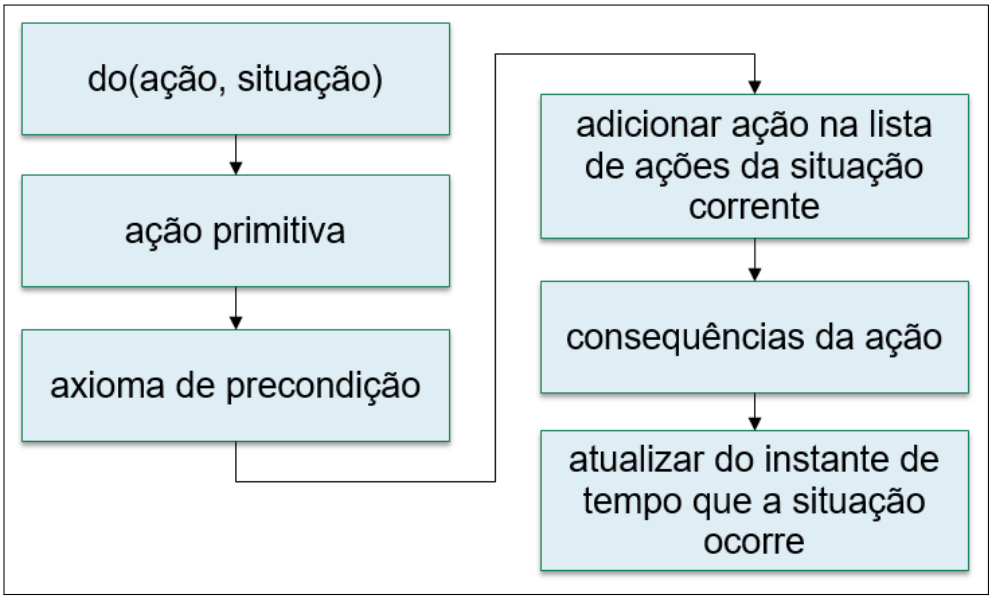
A seguir serão descritos detalhes sobre as linhas de código presentes no Quadro 1.

Na linha 2 a função *primitive_action(A)* verifica se a ação que está sendo executada é uma das ações primitivas previstas pelo sistema. As ações que podem ser executadas são: *startMove/4*, *endMove/4*, *startPan/2*, *endPan/2*, *sense/3*.

Na linha 3 a função *poss(A,S)* representa o axioma de precondição da ação que está sendo executada. Os respectivos axiomas foram descritos na Seção 3.2.2.

Na linha 5 a função *actionConsequences(A,S)* altera o estado do domínio de acordo com a ação que está sendo executada. Foi verificada uma necessidade de expressar consequências lógicas para as ações que acontecem no mundo devido a algumas percepções serem dependentes

Figura 11 – Estrutura axioma *do(a,s)*



Fonte: Autor

Quadro 1 – Axioma do(A,S)

```
do(A,S):-
    primitive_action(A),
    poss(A,S),
    start(S,T1),
    actionConsequences(A,S),
    update_current_situation(A),
    time(A,T),
    retract(start(S,T1)),
    assertz(start(S,T)).
```

Fonte: Autor

umas das outras. Essa função se assemelha às ações exógenas descritas em (GROSSKREUTZ; LAKEMEYER, 2001). Por exemplo, o fluente *facing/3* representa que o robô está posicionado de forma que seu sensor esteja direcionado γ graus em relação ao Norte no local $loc(x_r, y_r)$ na situação s. Na possibilidade do sistema possuir a informação de *facing/3* na sua base de conhecimento (situação s_0), e uma ação do tipo *endMove/4* referente à movimentação do robô for realizada, o local $loc(x_r,y_r)$ do fluente *facing/3* que está em memória estará defasado e assim se faz necessário sua atualização. A lista de consequências lógicas está descrita a seguir. Posteriormente no Capítulo 6 será discutido a independência do fluente *facing/3* das ações de movimento *startMove/4* e *endMove/4*.

a) *actionConsequences(startMove(Body,Location1,Location2,Time),S)*: não foram encontradas consequências lógicas que devessem estar explicitamente codificadas pois não ocorre mudança de localização efetiva através desta ação, ou seja, os axiomas de estado sucessor são suficientes para tratar a ação *startMove/4*;

Quadro 2 – Função *actionConsequences/2* para a ação *endPan/2*

```
1  actionConsequences(endPan(Omega,Time), Situation):-
2      robot(Robot),
3      location(Robot, Location, Situation),
4      facing(OldTheta, OldLocation, Situation),
5      OmegaT is (OldTheta+Omega) mod 360,
6      (
7         OmegaT >= 0,
8         OmegaT =< 180,
9         NewTheta is OmegaT, !;
10        OmegaT > 180,
11        NewTheta is (-1*(180 - (OmegaT - 180))), !;
12        OmegaT < 0,
13        OmegaT >= -180,
14        NewTheta is OmegaT, !;
15        OmegaT < -180,
16        NewTheta is (180 + (OmegaT + 180)), !
17     ),
18     retract(facing(OldTheta, OldLocation, Situation)),
19     assertz(facing(NewTheta, Location, Situation)).
```

Fonte: Autor

b) *actionConsequences(startPan(Omega,Time),S)*: similar à ação *startMove/4*;

c) *actionConsequences(endMove(Body,Location1,Location2,Time),S)*: devido ao fluente *location/3* estar armazenado em memória, se faz necessário que toda ação *endMove/4* atualize esta informação tanto para movimentos do robô quanto para movimentos dos objetos que se encontram em seu campo de visão. Adicionalmente, o fluente *facing/3* também necessita de uma atualização da informação que está em memória, pois seu segundo parâmetro é referente ao local em que o robô se encontra. Então, quando existe o término de movimento do robô, obrigatoriamente deve-se atualizar os fluentes *location/3* e *facing/3*;

d) *actionConsequences(endPan(Omega,Time),S)*: a ação *endPan* é responsável por informar ao sistema que o robô terminou um movimento de rotação de ângulo γ, que pode ser positivo representando que o sentido da rotação é horária, ou negativo representando sentido anti-horário. O fluente *facing/3* por sua vez representa o ângulo θ em que o robô está direcionado em relação ao Norte. Desta forma, se faz necessário calcular o ângulo de orientação do robô através de uma consequência lógica da ação *endPan/2*. No Quadro 2 pode-se visualizar a implementação em Prolog deste cálculo. Nesta figura, as linhas 6 a 17 efetuam uma verificação de quadrante do ângulo resultante da soma de θ com γ e normaliza de forma que um resultado positivo indique que o robô está direcionado θ graus no sentido horário

em relação ao Norte, ou em caso de resultado negativo o valor representa que o deslocamento é no sentido anti-horário.

e) *actionConsequences(sense(Profile,loc(Xr,Yr),Time),S)*: gera o perfil de profundidade *Profile* através dos sensores do robô e atualiza o valor armazenado na função *sensedProfile(P)* criada para armazenar em memória a última leitura dos sensores do robô móvel.

Na linha 6 do Quadro 1 a função *update_current_situation/1* atualiza a situação S com uma lista de ações atualizada, que contém a ação que está sendo realizada. Sua implementação em Prolog pode ser vista no Quadro 3. Nesta função, a linha 2 é responsável por remover a lista de ações As da situação $s0$ que estava armazenada em memória, e adicionar em seu lugar a lista atualizada $[A|As]$, através da linha 3.

Quadro 3 – Função *update_current_situation/1*

```
1 update_current_situation(A):-
2     retract(situation(s0,As)),
3     assertz(situation(s0,[A|As])).
```

Fonte: Autor

Na linha 9 do Quadro 1 a função *assertz(start(S,T))* atualiza o instante de tempo em que a situação S se encontra com o valor de T em que a ação é realizada no domínio.

4.1.2 Axiomas auxiliares

Quadro 4 – Função *euD*

```
1 euD(loc(X1, Y1), loc(X2, Y2), Distance):-
2     Z is (X1 - X2)^2 + (Y1 - Y2)^2,
3     Distance is sqrt(Z).
```

Fonte: Autor

Os predicados auxiliares desenvolvidos servem para facilitar os cálculos realizados pelos fluentes relacionais descritos no capítulo anterior. O mais simples é a função *euD/3* que é responsável pelo cálculo da distância Euclidiana entre dois locais. Sua implementação em Prolog pode ser verificada no Quadro 4.

O segundo axioma auxiliar que foi desenvolvido, que já foi descrito no Capítulo 3, é o $angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r),\gamma)$. Nesta função, através da informação de coordenadas de cada objeto em cena, a distância angular de ambos os locais pode ser calculada em Prolog conforme o Quadro 5. Para o cálculo são utilizados vetores que partem do local atual do robô para o local dos objetos $bodyA$ e $bodyB$ (linhas 2 e 3). Posteriormente, através de Geometria Analítica, o ângulo entre os dois vetores é calculado (linha 4). Por fim, na linha 5 obtém-se o valor absoluto do ângulo calculado. Diferentemente do axioma auxiliar *angle/6*, descrito em

Quadro 5 – Função *angleV*

```
1 angleV(loc(Xa, Ya), loc(Xb, Yb), loc(Xr, Yr), Gamma):-
2       Vax is Xa-Xr, Vay is Ya-Yr,
3       Vbx is Xb-Xr, Vby is Yb-Yr,
4       GammaT is acos( ( ( Vax * Vbx ) + ( Vay * Vby ) ) / ( sqrt( Vax^2 + Vay^2 ) *
    sqrt( Vbx^2 + Vby^2 ) ) ) * 180 / pi,
5       Gamma is abs(GammaT).
```

Fonte: Autor

Soutchanski e Santos (2008), o ângulo calculado por *angleV/4* não representa diretamente a distância angular entre dois objetos, pois não são considerados os raios, ou tamanhos aparentes de cada objeto. A responsabilidade de determinar a distância angular foi transferida para os axiomas relacionais que utilizam tal função quando necessário.

Quadro 6 – Funções *previous/3* e *prevAction/3*

```
1 prevAction(PreviousAction,Action,ActionList):-
2       ( var(Action)
3       -> [PreviousAction|_] = ActionList
4       ; append(_,[Action,PreviousAction|_],ActionList) ).
```

Fonte: Autor

Outra função auxiliar definida no sistema é a *prevAction/3*. A responsabilidade da função *prevAction/3* é responder qual é a ação $actionB$ que precedeu uma ação $actionA$ no histórico da situação $s0$. Sua implementação em Prolog pode ser vista no Quadro 6.

A função *prevAction/3* tem dois fluxos lógicos. Quando o segundo parâmetro é vazio, ou seja, quando não é passada uma ação de referência para que a função retorne a ação anterior, o resultado será a primeira ação da lista de ações $ActionList$. Por outro lado, quando existe uma ação de referência, a função retornará a ação executada imediatamente antes dela.

4.1.3 Fluentes

Nesta seção serão tratados todos os fluentes propriamente descritos no capítulo anterior que representam os atributos dos objetos, percepções de movimento e relações espaciais. A tradução dos axiomas para Prolog é simples e exemplificaremos através da descrição detalhada abaixo do predicado *extending/3*.

A percepção de que um pico de profundidade esteja crescendo (*extending/3*) é resultado de uma entre quatro possibilidades:

a) O robô móvel efetuou uma ação de sensoriamento e registrou através de seus sensores um perfil de profundidade que contém um pico que possui tamanho angular maior do que na situação anterior. Além disso, para que seja verdadeiro, é verificada a localização atual do robô na situação corrente para calcular se a disparidade

de distância entre objeto e robô é menor do que o limiar de profundidade L, conforme Quadro 7:

Quadro 7 – Axioma *extending/3* para ação de sensoriamento

```
extending(pk(Body, DepthA, SizeA, DistanceA), loc(Xr, Yr), do(A, S)):-
    A=sense(Profile, loc(Xr, Yr), Time),
    member(pk(Body, DepthB, SizeB, DistanceB), Profile),
    robot(Robot),
    location(Robot, loc(Xr, Yr), S),
    limitBoundary(L),
    size(pk(Body, DepthA, SizeA, DistanceA), SizeA, loc(Xr, Yr), S),
    depth(pk(Body, DepthA, SizeA, DistanceA), DepthA, loc(Xr, Yr), S),
    SizeA < SizeB,
    DepthA =< L,
    DepthB =< L, !.
```

Fonte: Autor

b) O robô móvel efetua uma movimentação de modo que a alteração de sua localização faça com que o tamanho angular calculado de um pico de profundidade que já estava em seu campo de visão fique maior (Quadro 8);

Quadro 8 – Axioma *extending/3* para movimentação do robô

```
extending(pk(Body, DepthA, SizeA, DistanceA), loc(Xa, Ya), do(A, S)):-
    A=startMove(Robot, loc(Xa, Ya), loc(Xr, Yr), Time),
    robot(Robot),
    location(Robot, loc(Xa, Ya), S),
    limitBoundary(L),
    location(Body, loc(Xb, Yb), S),
    radius(Body, Radius),
    euD(loc(Xb, Yb), loc(Xr, Yr), EuDistance),
    SinAlpha is Radius/(EuDistance),
    Alpha is asin(SinAlpha),
    SizeB is 2*Alpha*180/pi,
    DepthB is EuDistance - Radius,
    size(pk(Body, DepthA, SizeA, DistanceA), SizeA, loc(Xa, Ya), S),
    depth(pk(Body, DepthA, SizeA, DistanceA), DepthA, loc(Xa, Ya), S),
    SizeA < SizeB,
    DepthA =< L,
    DepthB =< L, !.
```

Fonte: Autor

c) Um objeto que estava no campo de visão do robô efetua uma movimentação de modo que a alteração de sua localização faça com que o tamanho angular calculado do seu pico de profundidade seja maior do que na situação anterior (Quadro 9);

Quadro 9 – Axioma *extending/3* para movimentação do objeto

```
extending(pk(Body, DepthA, SizeA, DistanceA), loc(Xr, Yr), do(A, S)):-
    A=startMove(Body, loc(Xa, Ya), loc(Xb, Yb), Time),
    robot(Robot),
    location(Robot, loc(Xr, Yr), S),
    limitBoundary(L),
    location(Body, loc(Xa, Ya), S),
    radius(Body, Radius),
    euD(loc(Xb, Yb), loc(Xr, Yr), EuDistance),
    SinAlpha is Radius/(EuDistance),
    Alpha is asin(SinAlpha),
    SizeB is 2*Alpha*180/pi,
    DepthB is EuDistance - Radius,
    size(pk(Body, DepthA, SizeA, DistanceA), SizeA, loc(Xr, Yr), S),
    depth(pk(Body, DepthA, SizeA, DistanceA), DepthA, loc(Xr, Yr), S),
    SizeA < SizeB,
    DepthA =< L,
    DepthB =< L, !.
```

Fonte: Autor

d) A última ação não tem influência nem na localização do robô, nem na localização do objeto em seu campo de visão, e tampouco foi uma ação de sensoriamento que altere a percepção de tamanho angular da situação anterior, mas que a percepção de crescimento do pico de profundidade do objeto continua a mesma (Quadro 10).

Neste momento deve-se chamar a atenção para um ponto importante da implementação. Em Soutchanski e Santos (2008) os axiomas de transição foram definidos com respeito à movimentação do robô móvel ou dos objetos em cena através da ação *endMove/3*. Entretanto, o entendimento que se tem é que, para que as transições *extending/3*, *shrinking/3*, *appearing/3*, *vanishing/3*, *approaching/4*, *receding/4*, *coalescing/4*, *splitting/4* e *hiding/4* sejam verdadeiras, é necessário que exista movimento. Mas se a ação *endMove/3* for considerada conforme é descrito no artigo, o movimento é na verdade encerrado. Por este motivo foi utilizada a ação *startMove/3* para determinar a existência de movimento e assim permitir a inferência destas transições. De modo contrário, a localização do robô e sua direção são sempre consideradas para cálculo no término do movimento.

De forma inversa, a ação *endMove/3* é fundamental para que os fluentes *peak_static/3* e *two_peak_static/4* sejam verdadeiros, pois para estes é necessário que não exista movimento. Assim, toda vez que os objetos em cena ou robô cessam movimento, e consequentemente não existe uma transição, os fluentes de pico estático são verdadeiros.

Um outro detalhe quanto à implementação é que, devido à utilização de recurso de atualização de dados em memória a respeito da situação $s0$, foi necessário escrever um novo axioma para cada fluente considerando a situação s, e não a situação $do(a,s)$. Este axioma veri-

Quadro 10 – Axioma *extending/3* para o problema do quadro

```
extending(pk(Body, DepthA, SizeA, DistanceA), loc(Xr, Yr), do(A, S)):-
    limitBoundary(L),
    not(( A=sense(Profile, loc(Xr, Yr), Time),
        member(pk(Body, DepthB, SizeB, _), Profile),
        ( SizeA >= SizeB, !;
          DepthB > L, ! ) )),
    not(A=endMove(Robot, loc(Xr, Yr), loc(Xrn, Yrn), Time)),
    not(A=endMove(Body, loc(Xb, Yb), loc(Xbn, Ybn), Time)),
    location(Robot, loc(Xr, Yr), S),
    location(Body, loc(Xb, Yb), S),
    situation(S,ActionList),
    prevAction(PreviousAction, A, ActionList),
    extending(pk(Body, DepthA, SizeA, DistanceA), loc(Xr, Yr),
do(PreviousAction, S)),
    !.
```

Fonte: Autor

fica através da função *situation/2* qual foi a última ação realizada e transfere a unificação para os axiomas descritos anteriormente. A implementação em Prolog pode ser vista no (Quadro 11).

Quadro 11 – Axioma *extending/3* na situação corrente

```
extending(pk(B, Depth, Size, Distance), loc(Xr, Yr), S):-
    situation(S,[A|_]),
    extending(pk(B, Depth, Size, Distance), loc(Xr, Yr), do(A, S)),
    !.
```

Fonte: Autor

Os demais axiomas de transição e de atributo são análogos quanto às notas de implementação descritas neste capítulo.

Os testes realizados foram estruturados ao redor de uma longa trajetória onde cada teste foi concentrado em trechos deste percurso, conforme apresentado no próximo capítulo.

4.2 MÉTODO SEM ATUALIZAÇÃO DE SITUAÇÃO EM MEMÓRIA

Este método contempla a implementação tradicional em Cálculo de Situações e não foi utilizado um recurso de armazenamento de situações em memória. Isto significa que o robô móvel é inicializado com uma base de conhecimento inicial do qual partem todos os cálculos de raciocínio espacial. O resultado destes cálculos e as ações que ocorrem no domínio permanecem disponíveis apenas em tempo de cálculo. Cada cálculo realizado remete à situação inicial do sistema.

A diferença principal do método anterior para este está na forma em que as ações são executadas. Nesta versão não existe um predicado responsável por gerenciar as ações executa-

das e atualizar a situação corrente. Cada ação executada é adicionada em um registrador que armazena a informação de situação desde o início do sistema. Este registrador é utilizado em todos os cálculos e é responsável por manter o controle do que ocorre no domínio do robô. Se o robô for reinicializado a informação é perdida e ele volta a ter apenas o conhecimento da situação inicial. Para que isto seja realizado, fui utilizada uma função *do* simplificada

$$do(A, S, do(A,S)),$$

que é utilizada no algoritmo de testes para execução de uma determinada lista de ações.

Exemplificando, caso se queira questionar o valor $size$ de um determinado corpo ao longo da execução do sistema, deverá ser feito uma consulta conforme (Quadro 12).

Quadro 12 – Processo de teste para pico individual

```
1 ?- forall(pstinitFluents(INIT),INIT),
2       size(pk(bodyA,_,S,_),S,loc(0.0,8.0),
3       do(startMove(cpto,loc(0.0,8.0),loc(1.83,7.93),20),
4             do(endMove(cpto,loc(0.1,8.1),loc(0.0,8.0),1),
5                   do(startMove(cpto,loc(0.1,8.1),loc(0.0,8.0),1),
6                         do(endMove(cpto,loc(0.0,8.0),loc(0.1,8.1),1),
7                               do(startMove(cpto,loc(0.0,8.0),loc(0.1,8.1),1),s0)))))).
8 S = 12.453700594268728.
```

Fonte: Autor

A consulta do (Quadro 12) se inicia com o predicado de alta ordem *forall* que inicia a situação inicial do sistema e define $s0$. Em seguida o fluente *size/4* é consultado de forma que são passados de parâmetro o corpo em que se deseja medir o tamanho angular $bodyA$, o local que representa o ponto de vista do robô $loc(0.0, 8.0)$ e a situação corrente através do cascateamento de todas as ações executadas no domínio desde a situação inicial.

Os fluentes de transição para picos de profundidade e pares de picos permanecem quase da mesma forma. A única exceção é o axioma da no (Quadro 11) que recebe a situação corrente como parâmetro, que foi desenvolvido exclusivamente para atender as necessidades do método da seção anterior, que passa a não ser mais necessário.

Os axiomas auxiliares *euD/3* e *angleV/4* foram reaproveitados sem modificação. Já as funções *update_current_situation/1* e *actionConsequences/2* não são necessárias neste método.

Os códigos utilizados neste trabalho de ambos os métodos estão disponíveis por completo através do endereço *https://github.com/pava197/qsrTdm-FEI*. Para execução, deve ser consultado apenas o módulo correspondente a qual método deseja-se testar por vez.

5 TESTES

Os testes realizados tiveram como objetivo observar o funcionamento dos axiomas codificados em Prolog em situações diversas, de modo que o resultado esperado fosse o mesmo do que foi proposto em (SOUTCHANSKI; SANTOS, 2008). Para a realização dos testes foram escritas sequências de ações que colocam o robô móvel em situações que o leve a observar todos os fluentes. A primeira fase dos testes realizados considera a base de dados inicial D_{S_0} a seguir:

a) $situation(s0,[])$: representa o conjunto de ações já realizadas na situação corrente, que inicialmente é uma lista vazia de ações;

b) $start(s0, 0)$: a situação inicial $s0$ se inicia no instante de tempo 0;

c) $location(cpto, loc(-13, -2), s0)$: o robô móvel denominado *cpto* está localizado inicialmente no local $loc(-13, -2)$;

d) $location(bodyA, loc(-7,2), s0)$: o objeto $bodyA$ está localizado inicialmente no local $loc(-7,2)$;

e) $location(bodyB, loc(5,1), s0)$: o objeto $bodyB$ está localizado inicialmente no local $loc(5,1)$;

f) $facing(90, loc(-13, -2), s0)$: o robô $cpto$ está inicialmente orientado à noventa graus no sentido horário em relação ao Norte;

g) $fieldView(120)$: o ângulo de abertura dos sensores do robô é de cento e vinte graus;

h) $limitBoundary(15)$: o alcance de profundidade dos sensores do robô é de quinze unidades;

i) $radius(bodyA,1)$: o raio do objeto $bodyA$ é de uma unidade;

j) $radius(bodyB,2)$: o raio do objeto $bodyB$ é de duas unidades;

k) $radius(cpto,0)$: o raio do objeto $cpto$ é considerado como zero unidades;

l) $thresholdBoundary(16)$: limiar de profundidade relacionado ao alcance dos sensores do robô;

m) $thresholdUnion(10)$: limiar de distância relacionado à transição de pares de picos.

Além disso, foi considerado como procedimento inicial do sistema adicionar um movimento do robô para que o sistema alimente a lista de ações da situação inicial. Isto foi realizado

com a intenção de inicializar os valores dos atributos de tamanho angular, disparidade e profundidade dos objetos que estiverem em cena. Ao utilizar este artifício, os axiomas tornam-se capazes de trabalhar como esperado pois podem obter o resultado através do raciocínio sobre as ações realizadas. Caso contrário, todos os atributos deveriam ser inicializados através de uma ação *sense/3*, e como este trabalho foi baseado em um simulação, os valores seriam manualmente inseridos em D_{S_0}.

5.1 TESTES DE MOVIMENTAÇÃO

Nesta seção serão descritos os testes realizados referentes aos fluentes *moving/4*, *rotating/2*, *facing/3*, *location/3*. Esta bateria de testes foi realizada utilizando o método com armazenamento da situação corrente em memória. Assumindo-se que o robô móvel *cpto* está localizado em $loc(x_{ra},y_{ra})$ na situação s, este efetua um movimento que o leva para o local $loc(x_{rb},y_{rb})$, e efetua em sequência um movimento de rotação que altera o ângulo θ em que o seu sensor está posicionado. Espera-se que após iniciar a ação de movimento, o fluente *moving/4* torne-se verdadeiro, da mesma forma que é esperado que após o início de rotação do sensor do robô, o fluente *rotating/2* também torne-se verdadeiro. Ambos os fluentes devem se tornar falsos quando ocorrer as ações *endMove/4* e *endPan/2*.

Quadro 13 – Teste de precondição de movimento e rotação

```
1 ?- poss(startMove(cpto, loc(-13,-2), loc(-12,-2), 1), s0).
2       true.
3 ?- poss(startPan(-15, 1), s0).
4       true.
```

Fonte: Autor

Antes das ações serem executadas no mundo, pode-se verificar a possibilidade de ação do robô através dos axiomas de precondição descritos no Capítulo 3. Nesta teoria porém não foi definida uma precondição de obstrução de rota ou destino. Por este motivo o sistema irá responder verdadeiro para a possibilidade de se mover o robô para um local que já esteja ocupado por outro corpo. Por outro lado, caso esta funcionalidade fosse ser adicionada ao sistema, deverá ser desenvolvida dentro dos axiomas de precondição de cada ação primitiva. De acordo com a situação ilustrada no Quadro 13, a possibilidade do robô localizado em $loc(-13, -2)$ se movimentar para o local $loc(-12, -2)$ é definida por $poss(startMove(cpto, loc(-13, -2), loc(-12, -2), 1), s_0)$. De forma similar, a possibilidade do robô rotacionar o seu sensor em quinze graus no sentido anti-horário é dada por $poss(startPan(-15, 1), s_0)$. Ao validar ambos os predicados na implementação em Prolog, obtêm-se o resultado *true*, conforme Quadro 13.

Na sequência, o robô *cpto* inicia o movimento utilizando o predicado $do(startMove(cpto, loc(-13, -2), loc(-12, -2), 1), s_0)$, e inicia a rotação do sensor com o predicado $do(startPan(-15, 1), s_0)$, por fim o valor dos fluentes na situação atual é testado. O fluente *moving/4* (linha 9) indica que o robô *cpto* está em movimento na situação s_0, partindo

Quadro 14 – Início de movimento e rotação

```
1  ?- location(cpto, Location, s0).
2        Location = loc(-13, -2).
3  ?- facing(Theta,loc(-13,-2),s0).
4        Theta = 90.
5  ?- do(startMove(cpto, loc(-13,-2), loc(-12,-2), 1), s0).
6        true.
7  ?- do(startPan(-15, 1), s0).
8        true.
9  ?- moving(cpto, loc(-13,-2), loc(-12,-2), s0).
10       true.
11 ?- rotating(-15,s0).
12       true.
```

Fonte: Autor

do local $loc(-13, -2)$ com destino ao local $loc(-12, -2)$. Já o fluente *rotating/2* (linha 11) indica que o robô está efetuando uma rotação de ângulo γ de quinze graus no sentido anti-horário. Neste ponto, caso seja consultado o axioma $facing(Theta, L, s01)$ o sistema retornará o ângulo e local da situação inicial. Isto se deve ao fato da teoria considerar a conclusão de movimento e de rotação para que os valores de localidade e ângulo θ possam ser atualizados. Para tanto, é utilizada a premissa de que cada ação é curta o suficiente para não interferir na continuidade de uma sequência de ações dentro do domínio.

Quadro 15 – Fim de movimento e rotação

```
1  ?- do(endMove(cpto, loc(-13,-2), loc(-12,-2), 1), s0).
2        true.
3  ?- do(endPan(-15, 1), s0).
4        true.
5  ?- moving(cpto, loc(-13,-2), loc(-12,-2), s0).
6        false.
7  ?- rotating(-15,s0).
8        false
9  ?- location(cpto, Location, s0).
10       Location = loc(-12, -2).
11 ?- facing(Theta,loc(-12,-2),s0).
12       Theta = 75.
```

Fonte: Autor

No Quadro 15 pode-se verificar que após as finalizações de movimento e de rotação através de *endMove/4* e *endPan/2*, os fluentes *moving/4* e *rotating/2* tornaram-se falsos. Vale notar que o fluente *facing/3* foi atualizado para a nova localização do robô $loc(-12, -2)$, verificada pelo fluente *location/3*, e que o sistema retornou na variável *Theta* o ângulo atualizado em relação no Norte.

5.2 TESTE SOBRE ATRIBUTOS

Utilizando a mesma sequência de ações descrita na situação da seção anterior, pode-se verificar que as ações possuem consequências lógicas sobre a percepção que o robô tem sobre o mundo e que estas podem ser observadas por meio dos fluentes *size/4*, *depth/4* e *dist/4*. Antes da execução das ações do Quadro 14, o robô possui a informação de que o objeto $bodyA$, que está inicialmente em seu campo de visão, possui tamanho angular z de valor $15{,}94$ graus, profundidade u de valor $6{,}21$ unidades e distância angular d da borda esquerda do campo de visão do robô no valor de $18{,}34$ graus, que pode ser verificado conforme Quadro 16.

Quadro 16 – z, u e d antes das ações de movimento

```
1 ?- visible(loc(-13,-2), bodyA, 120, 90, s0).
2      true.
3 ?- size(pk(bodyA, _, Size, _), Size, loc(-13,-2), s0).
4      Size = 15.94.
5 ?- depth(pk(bodyA, Depth, _, _), Depth, loc(-13,-2), s0).
6      Depth = 6.21.
7 ?- dist(pk(bodyA , _, _, Distance ), Distance, loc(-13,-2), s0).
8      Distance = 18.34.
```

Fonte: Autor

Após o robô se movimentar para $loc(x_{rb}, y_{rb})$ e terminar a rotação do seu sensor, os fluentes referentes aos atributos do objeto $bodyA$ foram testados na expectativa de se observar mudança de valores.

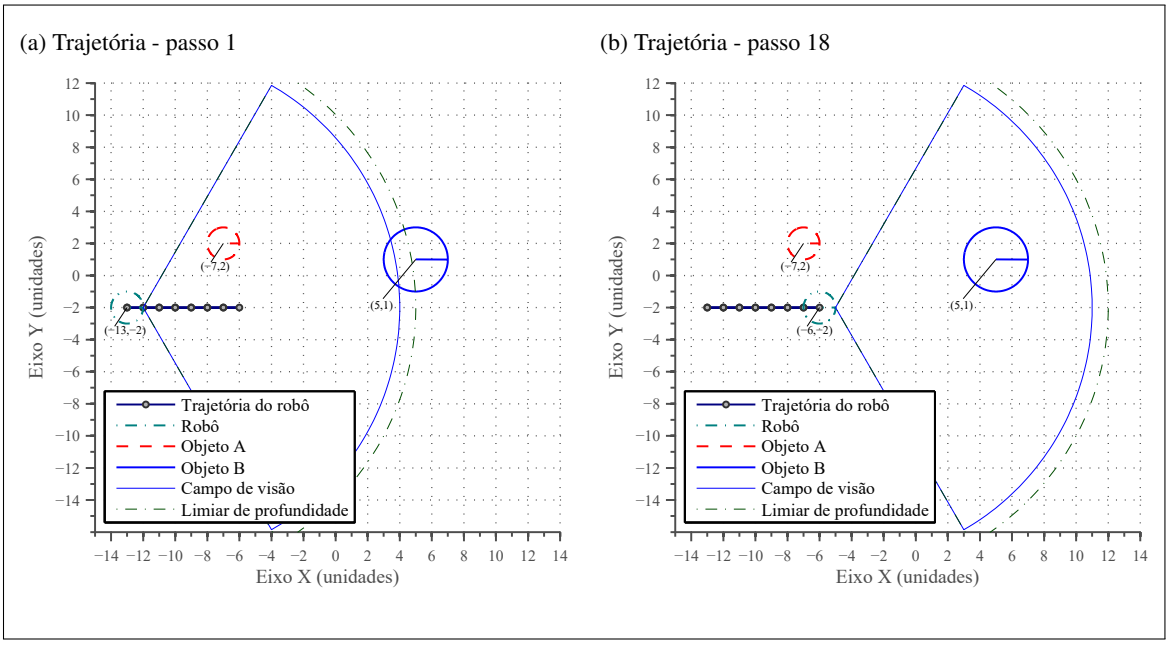
Quadro 17 – z, u e d após ações de movimento

```
1 ?- visible(loc(-12,-2), bodyA, 120, 75, s0).
2      true.
3 ?- size(pk(bodyA, _, Size, _), Size, loc(-12,-2), s0).
4      Size = 17.97.
5 ?- depth(pk(bodyA, Depth, _, _), Depth, loc(-12,-2), s0).
6      Depth = 5.40.
7 ?- dist(pk(bodyA , _, _, Distance ), Distance, loc(-12,-2), s0).
8      Distance = 27.35.
```

Fonte: Autor

Pode-se observar que o tamanho angular z, representado pela variável *Size* no código, aumentou naturalmente para $17{,}97$ graus conforme o robô se aproximou do objeto. Da mesma forma que a disparidade de distância entre o robô e o objeto diminuiu para $5{,}40$ unidades. Foi observado também que a distância angular do objeto aumentou para $27{,}35$ graus visto que o robô efetuou uma rotação de quinze graus no sentido anti-horário, o que fez com que o objeto $bodyA$ se afastasse da borda esquerda do campo de visão do robô. Outra validação que pode-se realizar é sobre a visibilidade de objeto $bodyA$ ao longo do tempo, através do Axioma *visible/5*, que permaneceu verdadeiro durante o teste.

Figura 12 – Trajetória de testes para transições de pico individual



Fonte: Autor

Com este teste podemos afirmar que a linguagem de ações do sistema reflete diretamente a representação e percepção espacial do robô móvel, quanto aos objetos que se encontram em seu campo de visão.

5.3 TESTES DE TRANSIÇÃO PARA PICO DE PROFUNDIDADE

O próximo teste refere-se às transições para picos individuais. Assume-se uma sequência de ações que aproxime e afaste o robô *cpto* do objeto *bodyA*, de modo que permita que se teste os Axiomas *appearing/3*, *extending/3*, *shrinking/3*, *vanishing/3* e *peak_static/3*. No primeiro momento foram testadas as transições do objeto levando em consideração o alcance de profundidade do sensor do robô, definido por L. Na Figura 12a pode-se observar uma ilustração do início do teste, e na Figura 12b o ponto de retorno do robô, de onde ele começa a se afastar do objeto $bodyA$.

5.3.1 Processo de testes de pico simples

Para que as ações fossem executadas de forma contínua, e não houvesse erros manuais de coleta de dados dos testes, foi escrito um predicado conforme Quadro 18. Na linha 2, o predicado *initsingle* representa toda a informação da situação inicial do sistema que é carregada em memória inicialmente. Na linha 4, todas as ações do teste que foram recuperadas pelo predicado *actionsingle* são executadas, e seus resultados em caso de verdadeiro ou falso são impressos na linha 5. Em seguida na linha 7 os fluentes referentes aos atributos são testados, e finalmente na linha 8 os fluentes relacionais para picos individuais são testados. O predi-

Quadro 18 – Processo de teste para pico individual

```
1 test_single_peak :-
2       forall(initsingle(INIT),INIT),
3       forall(actionsingle(DO),
4       ( DO,
5          writeln(actionsingle:DO),
6          location(cpto, loc(X,Y), s0),
7          forall(querysingleAttribute(X,Y,AT), (AT, writeln(AT))),
8          forall(querysingle(X,Y,Q), (Q -> writeln(true:Q) ; writeln(false:Q)))
9       )).
```

Fonte: Autor

cado $querysingleAttribute$ é responsável pela consulta dos fluentes *depth/4*, *size/4* e *dist/4* após cada ação executada, e o predicado querysingle é responsável pela consulta dos fluentes de transições para pico individual *peak_static/3*, *appearing/3*, *extending/3*, *shrinking/3* e *vanishing/3*.

5.3.2 Teste do fluente *appearing/3*

Esta subseção tem o objetivo de testar o fluente *appearing/3* e compreende uma sequência de ações de movimento do robô móvel que o levam de uma posição na qual não é possível observar nenhum corpo, para uma posição em que um corpo entra em seu campo de visão. O teste se inicia com o robô móvel posicionado no local $loc(-13, -2)$ e se movimenta para o local $loc(-12, -2)$. Esta movimentação altera a percepção de distância que o robô tem do objeto observado no local $loc(5,1)$, e o valor de disparidade passa de 17.25 para 16.26, que ainda é superior ao limiar máximo de alcance dos sensores do robô que vale 16 unidades. O robô executa em sequência um novo movimento do local $loc(-12, -2)$ para o local $loc(-11, -2)$ que leva a disparidade de distância do pico de profundidade observado para o valor de 15.28 unidades. Este valor consta entre o limite nominal dos sensores que vale 15 unidades e o limiar máximo de distância de 16 unidades. Esta situação torna o fluente *appearing/3* verdadeiro, pois de forma correta representa que o corpo $bodyB$ que não podia ser observado passou a ser visível do ponto de vista do robô móvel. A sequência de ações executada está descrita no Quadro 19.

Quadro 19 – Sequência de ações para teste do fluente *appearing/3*

```
1 ?- do(startMove(cpto, loc(-13,-2), loc(-12,-2), 1), s0),
2       do(endMove(cpto, loc(-13,-2), loc(-12,-2), 2), s0),
3             do(startMove(cpto, loc(-12,-2), loc(-11,-2), 3), s0),
4                   appearing(pk(bodyB, Depth, Size, Distance), loc(-12,-2), s0).
5 Depth = 15.28, Size = 7.04.
6 ?- do(endMove(cpto, loc(-12,-2), loc(-11,-2), 4), s0).
7 true.
```

Fonte: Autor

5.3.3 Teste do fluente *extending/3*

Esta subseção tem o objetivo de testar o fluente *extending/3* e compreende uma sequência de ações de movimento do robô móvel que o levam de uma posição na qual ele observa um corpo que acaba de entrar em seu campo de visão, para uma posição em que o corpo está totalmente dentro de seu campo de visão. Neste teste, conforme o robô se aproxima do objeto observado, o tamanho angular do pico de profundidade que representa o objeto aumenta e a disparidade de distância entre o robô e o pico diminui. O teste se inicia com o robô posicionado no local $loc(-10, -2)$ e executa uma sequência de ações que o movimentam pra o local $loc(-9, -2)$. Ao realizar esta movimentação, o tamanho angular do pico de profundidade do objeto $bodyB$ aumenta de 7.50 para 8.01 graus, e a profundidade do pico passa de 14.30 para 13.32 unidades. Estes valores são menores do que o limite nominal de alcance dos sensores do robô que vale 15 unidades. Desta forma, o fluente *extending/4* torna-se verdadeiro a partir do instante em que ação de início de movimento é iniciada. Em seguida, foram realizadas novas movimentações do robô móvel em direção ao corpo $bodyB$, onde a cada início de movimento houve o teste do fluente *extending/4* que provou ser corretamente verdadeiro. A sequência de ações executadas neste teste está descrita no Quadro 20.

Quadro 20 – Sequência de ações para teste do fluente *extending/3*

```
1  ?- do(startMove(cpto, loc(-10,-2), loc(-9,-2), 7), s0),
2        extending(pk(bodyB, Depth, Size, Distance), loc(-10,-2), s0).
3  Depth = 14.30, Size = 7.50.
4  ?- do(endMove(cpto, loc(-10,-2), loc(-9,-2), 8), s0),
5        do(startMove(cpto, loc(-9,-2), loc(-8,-2), 9), s0),
6              extending(pk(bodyB, Depth, Size, Distance), loc(-9,-2), s0).
7  Depth = 13.32, Size = 8.01.
8  ?- do(endMove(cpto, loc(-9,-2), loc(-8,-2), 10), s0),
9        do(startMove(cpto, loc(-8,-2), loc(-7,-2), 11), s0),
10             extending(pk(bodyB, Depth, Size, Distance), loc(-8,-2), s0).
11 Depth = 12.34, Size = 8.60.
12 ?- do(endMove(cpto, loc(-8,-2), loc(-7,-2), 12), s0),
13       do(startMove(cpto, loc(-7,-2), loc(-6,-2), 13), s0),
14             extending(pk(bodyB, Depth, Size, Distance), loc(-7, -2), s0).
15 Depth = 11.37, Size = 9.27.
16 ?- do(endMove(cpto, loc(-7,-2), loc(-6,-2), 14), s0).
17 true.
```

Fonte: Autor

5.3.4 Teste do fluente *shrinking/3*

Esta subseção tem o objetivo de testar o fluente *shrinking/3* e compreende uma sequência de ações de movimento do robô móvel que o levam de uma posição na qual existe um corpo

dentro do seu campo de visão, para uma posição em que o corpo continua dentro do seu campo de visão porém mais distante. Neste teste, conforme o robô se afasta do objeto observado, o tamanho angular do pico de profundidade que representa o objeto diminui e a disparidade de distância entre o robô e o pico aumenta. O teste se inicia com o robô posicionado no local $loc(-6, -2)$ e efetua uma sequência de ações que o movimentam para o local $loc(-7, -2)$. Estas ações fazem com que o tamanho angular do pico de profundidade do corpo $bodyB$ passe de 10.06 para 9.27 graus, e a profundidade do pico se altera de 10.40 para 11.37 unidades. Após isso, são executadas novas ações de movimento que continuam a afastar o robô móvel do objeto observado. No início de cada passo de movimentação o fluente *shrinking/3* foi testado e provou ser verdadeiro de forma esperada, pois representa que um pico de profundidade diminuiu de tamanho enquanto o robô estava em movimento. A sequência de ações executadas neste teste está descrita no Quadro 21.

Quadro 21 – Sequência de ações para teste do fluente *shrinking/3*

```
1  ?- do(startMove(cpto, loc(-6,-2), loc(-7,-2), 19), s0),
2        shrinking(pk(bodyB, Depth, Size, Distance), loc(-6, -2), s0).
3  Depth = 10.40, Size = 10.06.
4  ?- do(endMove(cpto, loc(-6,-2), loc(-7,-2), 20), s0),
5        do(startMove(cpto, loc(-7,-2), loc(-8,-2), 21), s0),
6              shrinking(pk(bodyB, Depth, Size, Distance), loc(-7, -2), s0).
7  Depth = 11.37, Size = 9.27.
8  ?- do(endMove(cpto, loc(-7,-2), loc(-8,-2), 22), s0),
9        do(startMove(cpto, loc(-8,-2), loc(-9,-2), 23), s0),
10             shrinking(pk(bodyB, Depth, Size, Distance), loc(-8, -2), s0).
11 Depth = 12.34, Size = 8.60.
12 ?- do(endMove(cpto, loc(-8,-2), loc(-9,-2), 24), s0),
13       do(startMove(cpto, loc(-9,-2), loc(-10,-2), 25), s0),
14             shrinking(pk(bodyB, Depth, Size, Distance), loc(-9, -2), s0).
15 Depth = 13.32, Size = 8.01.
16 ?- do(endMove(cpto, loc(-9,-2), loc(-10,-2), 26), s0).
17 true.
```

Fonte: Autor

5.3.5 Teste do fluente *vanishing/3*

Esta subseção tem o objetivo de testar o fluente *vanishing/3* e compreende uma sequência de ações de movimento do robô móvel que o levam de uma posição na qual existe um corpo que está dentro do seu campo de visão, para uma posição em que o corpo ultrapassa o alcance nominal dos sensores do robô móvel configurado em 15 unidades, mas permanece dentro do limiar máximo de profundidade de 16 unidades. Neste teste, conforme o robô se afasta do objeto observado o tamanho angular do pico de profundidade que representa o corpo $bodyB$ diminui, e a disparidade de distância entre o robô e o objeto aumenta de forma que o pico de

profundidade entra numa região em que considera-se que o mesmo irá desaparecer do campo de visão do robô. O teste se inicia com o robô localizado em $loc(-10, -2)$. São executadas em seguida ações de movimento que levam o robô para a posição $loc(-11, -2)$. Estas alterações no domínio fazem com que o tamanho angular do pico de profundidade se altere de 7.50 para 7.04 graus, e a profundidade do pico passe de 14.30 para 15.28 unidades. Com isso, o fluente *vanishing/3* é testado, e de forma esperada apresentou o valor verdadeiro. Posteriormente o robô executa nova sequência de ações que o afastam ainda mais do objeto observado de modo que a distância entre os corpos se torna maior do que o alcance dos sensores o robô. Por este motivo, ou seja, pela incapacidade de se observar o corpo $bodyB$, o fluente tornou-se falso. A sequência de ações executadas no teste está descrita no Quadro 22.

Quadro 22 – Sequência de ações para teste do fluente *vanishing/3*

```
1  ?- do(startMove(cpto, loc(-10,-2), loc(-11,-2), 27), s0),
2        vanishing(pk(bodyB, Depth, Size, Distance), loc(-10, -2), s0).
3  Depth = 14.30, Size = 7.50.
4  ?- do(endMove(cpto, loc(-10,-2), loc(-11,-2), 28), s0),
5        do(startMove(cpto, loc(-11,-2), loc(-12,-2), 29), s0),
6              vanishing(pk(bodyB, Depth, Size, Distance), loc(-11, -2), s0).
7  false.
8  ?- do(endMove(cpto, loc(-11,-2), loc(-12,-2), 30), s0),
9        do(startMove(cpto, loc(-12,-2), loc(-13,-2), 31), s0),
10             vanishing(pk(bodyB, Depth, Size, Distance), loc(-12, -2), s0).
11 false.
```

Fonte: Autor

5.3.6 Teste do fluente *peak_static/3*

Esta subseção não possui uma sequência própria de ações. Isto se deve ao fato de que em todos os momentos em que o robô efetuou a ação *endMove* nos testes anteriores o fluente *peak_static/3* tornou-se verdadeiro. Isto se deve ao fato de que em cada uma dessas situações configurou-se um término de movimento. O Quadro 23 demonstra tal afirmação utilizando as ações do teste do fluente *appearing/3*.

5.3.7 Resultado dos testes de transição para pico de profundidade

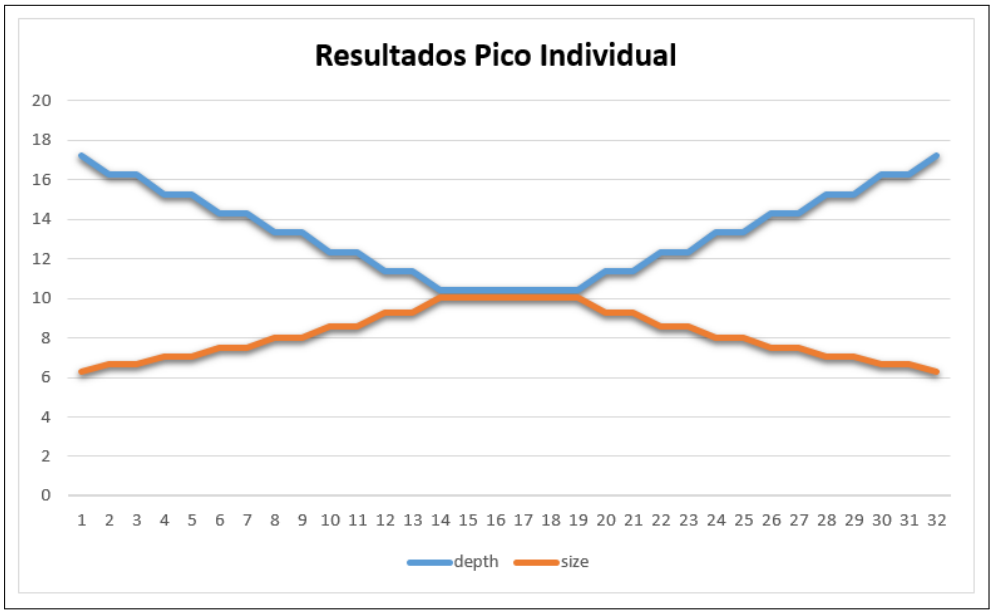
Todos os testes de transições para picos individuais de profundidade foram realizados com sucesso e todos os fluentes *appearing/3*, *extending/3*, *shrinking/3*, *vanishing/3* e *peak_static/3* foram verdadeiros nos momentos esperados. Na Figura (13) consta uma observação da relação entre o tamanho angular do pico de profundidade de um objeto observado e a disparidade de distância entre robô e objeto.

Quadro 23 – Sequência de ações para teste do fluente *peak_static/3*

```
1 ?- do(startMove(cpto, loc(-13,-2), loc(-12,-2), 1), s0),
2        do(endMove(cpto, loc(-13,-2), loc(-12,-2), 2), s0),
3              peak_static(pk(bodyB, Depth, Size, Distance), loc(-12,-2), s0).
4 Depth = 16.26, Size = 6.64, Distance = 46.67.
5 ?- do(startMove(cpto, loc(-12,-2), loc(-11,-2), 3), s0),
6        do(endMove(cpto, loc(-12,-2), loc(-11,-2), 4), s0),
7              peak_static(pk(bodyB, Depth, Size, Distance), loc(-11,-2), s0).
8 Depth = 15.28, Size = 7.04, Distance = 45.86.
```

Fonte: Autor

Figura 13 – Comparação de tamanho angular e disparidade de distância



Fonte: Autor

Os resultados de execução das ações de todos os testes para picos individuais de profundidade podem ser observados na Tabela 1.

Na linha 1 da Tabela 1 todos os fluentes de transição são falsos porque o objeto $bodyB$ está fora do alcance dos sensores do robô. Conforme o robô executa ações de movimento que o aproximam do objeto, o fluente *appearing/3* torna-se verdeiro após a ação número 3. É importante notar que o valor de profundidade u medido neste instante é de $16{,}263$, que é superior ao alcance máximo do sensor do robô adicionado ao limiar de profundidade de valor 16. No entanto, a ação $startMove(cpto,loc(-12,-2),loc(-11,-2),3)$ que o robô iniciou, descreve um movimento que será finalizado no local $loc(-11,-2)$, de onde pode-se calcular uma profundidade de pico de valor $15{,}279$. Por esta razão, nesta situação os fluentes *moving/4* e *appearing/3* são verdadeiros.

Posteriormente, na ação de número 7, o fluente *appearing/3* passa a ser falso e o fluente *extending/3* se torna verdadeiro. Isto é devido ao objeto $bodyB$ já constar no campo de visão do robô na situação anterior, e a disparidade de profundidade do objeto continuou a diminuir.

Tabela 1 – Resultados dos testes de transição de pico individual

action	#	depth	size	dist	peak_static	appearing	extending	shrinking	vanishing
do(startMove(cpto,loc(-13,-2),loc(-12,-2),1),s0)	1	17.248	6.283	47.396	false	false	false	false	false
do(endMove(cpto,loc(-13,-2),loc(-12,-2),2),s0)	2	16.263	6.642	46.671	**true**	false	false	false	false
do(startMove(cpto,loc(-12,-2),loc(-11,-2),3),s0)	3	16.263	6.642	46.671	false	**true**	false	false	false
do(endMove(cpto,loc(-12,-2),loc(-11,-2),4),s0)	4	15.279	7.044	45.858	**true**	false	false	false	false
do(startMove(cpto,loc(-11,-2),loc(-10,-2),5),s0)	5	15.279	7.044	45.858	false	**true**	false	false	false
do(endMove(cpto,loc(-11,-2),loc(-10,-2),6),s0)	6	14.297	7.496	44.942	**true**	false	false	false	false
do(startMove(cpto,loc(-10,-2),loc(-9,-2),7),s0)	7	14.297	7.496	44.942	false	false	**true**	false	false
do(endMove(cpto,loc(-10,-2),loc(-9,-2),8),s0)	8	13.318	8.010	43.900	**true**	false	false	false	false
do(startMove(cpto,loc(-9,-2),loc(-8,-2),9),s0)	9	13.318	8.010	43.900	false	false	**true**	false	false
do(endMove(cpto,loc(-9,-2),loc(-8,-2),10),s0)	10	12.342	8.597	42.707	**true**	false	false	false	false
do(startMove(cpto,loc(-8,-2),loc(-7,-2),11),s0)	11	12.342	8.597	42.707	false	false	**true**	false	false
do(endMove(cpto,loc(-8,-2),loc(-7,-2),12),s0)	12	11.369	9.274	41.327	**true**	false	false	false	false
do(startMove(cpto,loc(-7,-2),loc(-6,-2),13),s0)	13	11.369	9.274	41.327	false	false	**true**	false	false
do(endMove(cpto,loc(-7,-2),loc(-6,-2),14),s0)	14	10.402	10.063	39.713	**true**	false	false	false	false
do(startPan(1,15),s0)	15	10.402	10.063	39.713	**true**	false	false	false	false
do(endPan(1,16),s0)	16	10.402	10.063	38.713	**true**	false	false	false	false
do(startPan(-1,17),s0)	17	10.402	10.063	38.713	**true**	false	false	false	false
do(endPan(-1,18),s0)	18	10.402	10.063	39.713	**true**	false	false	false	false
do(startMove(cpto,loc(-6,-2),loc(-7,-2),19),s0)	19	10.402	10.063	39.713	false	false	false	**true**	false
do(endMove(cpto,loc(-6,-2),loc(-7,-2),20),s0)	20	11.369	9.274	41.327	**true**	false	false	false	false
do(startMove(cpto,loc(-7,-2),loc(-8,-2),21),s0)	21	11.369	9.274	41.327	false	false	false	**true**	false
do(endMove(cpto,loc(-7,-2),loc(-8,-2),22),s0)	22	12.342	8.597	42.707	**true**	false	false	false	false
do(startMove(cpto,loc(-8,-2),loc(-9,-2),23),s0)	23	12.342	8.597	42.707	false	false	false	**true**	false
do(endMove(cpto,loc(-8,-2),loc(-9,-2),24),s0)	24	13.318	8.010	43.900	**true**	false	false	false	false
do(startMove(cpto,loc(-9,-2),loc(-10,-2),25),s0)	25	13.318	8.010	43.900	false	false	false	**true**	false
do(endMove(cpto,loc(-9,-2),loc(-10,-2),26),s0)	26	14.297	7.496	44.942	**true**	false	false	false	false
do(startMove(cpto,loc(-10,-2),loc(-11,-2),27),s0)	27	14.297	7.496	44.942	false	false	false	false	**true**
do(endMove(cpto,loc(-10,-2),loc(-11,-2),28),s0)	28	15.279	7.044	45.858	**true**	false	false	false	false
do(startMove(cpto,loc(-11,-2),loc(-12,-2),29),s0)	29	15.279	7.044	45.858	false	false	false	false	false
do(endMove(cpto,loc(-11,-2),loc(-12,-2),30),s0)	30	16.263	6.642	46.671	**true**	false	false	false	false
do(startMove(cpto,loc(-12,-2),loc(-13,-2),31),s0)	31	16.263	6.642	46.671	false	false	false	false	false
do(endMove(cpto,loc(-12,-2),loc(-13,-2),32),s0)	32	17.248	6.283	47.396	**true**	false	false	false	false

Fonte: Autor

Após a ação de número 19, o robô começa a retornar à posição inicial, e por consequência se afasta do objeto $bodyB$. Deste modo, o fluente que se torna verdadeiro é o *shrinking/3*.

Na sequência, quando a ação $startMove(cpto,loc(-10,-2),loc(-11,-2),27)$ de número 27 é executada, o local de destino de robô será $loc(-11,-2)$ de onde pode-se calcular uma disparidade de profundidade u de valor 15.279. Este valor é maior do que o alcance dos sensores do robô (15), o que significa que o objeto $bodyB$ está na iminência de sair do campo de visão do robô, e por este motivo o fluente *vanishing/3* é verdadeiro.

Finalmente, o robô se afasta tanto do objeto $bodyB$ que seus sensores não conseguem mais gerar um perfil de profundidade que aponte a presença do objeto em seu campo de visão. Assim, os fluentes se tornam todos falsos com exceção do fluente *peak_static/3* que continuou verdadeiro cada vez que o robô finaliza um movimento.

5.4 TESTES DE TRANSIÇÃO PARA PARES DE PICOS

Nesta seção serão descritos os testes realizados para os fluentes relacionados aos pares de picos de profundidade. Assume-se que uma determinada sequência de ações aproxime e afaste dois picos de profundidade que representam os objetos $bodyA$ e $bodyB$ que estão no campo de visão do robô móvel $cpto$, de forma que seja possível testar os Axiomas *approaching/4*, *receding/4*, *coalescing/4*, *splitting/4*, *two_peak_static/4* e *hiding/4*. Este bateria de testes foi realizada utilizando o método sem armazenamento da situação corrente em memória, ou seja, sem a utilização do functor *assertz/1*.

5.4.1 Processo de testes para pares de picos

De forma similar ao que foi feito no teste anterior, nesta etapa também foi utilizado um comando para execução das ações dos testes de forma sequencial, conforme Quadro 24. A única diferença é que foram utilizadas listas de ações, fluentes e situação inicial diferentes, para que as informações pertinentes a cada transição espacial fosse testada.

Quadro 24 – Processo de teste para pares de picos

```
1  test_actions_pst :-
2    working_directory(_, 'C:/qsrTdm-FEI'),
3    (exists_file('C:/qsrTdm-FEI/test_actions_pst.txt') ->
   delete_file('C:/qsrTdm-FEI/test_actions_pst.txt') ; true),
4    (exists_file('C:/qsrTdm-FEI/test_actions_reverse_pst.txt') ->
   delete_file('C:/qsrTdm-FEI/test_actions_reverse_pst.txt'); true),
5    forall(pstinitFluents(INIT),INIT),
6    do(endMove(cpto, loc(0.1,8.1), loc(0.0,8.0), 1),
7       do(startMove(cpto, loc(0.1,8.1), loc(0.0,8.0), 1),
8          do(endMove(cpto, loc(0.0,8.0), loc(0.1,8.1), 1),
9             do(startMove(cpto, loc(0.0,8.0), loc(0.1,8.1), 1), s0))), S0),
10   pstactions(Actions),
11   act(Actions, S0,'test_actions_pst.txt'),
12   pstactions_reverse(ActionsReverse),
13   act(ActionsReverse, S0,'test_actions_reverse_pst.txt').
```

Fonte: Autor

Na Linha (5) do Quadro 24 o predicado *forall/2* inicia as informações da situação inicial. Em seguida, são adicionadas 4 acões iniciais conforme Linhas (6), (7), (8) e (9). As funções *pstactions* e *pstactions_reverse/2* retornam listas de ações que serão analisadas com o predicado

act/3 no Quadro 25. Além disso, o predicado *test_actions_pst* do Quadro 24 reinicia os logs de saída do sistema para um registro de execução limpo.

Quadro 25 – Predicado *act/3*

```
1 act([],_,_).
2 act([LH|LT],S0,Filename):-
3    do(LH,S0,S2),
4    writeln(do(LH,S0),Filename),
5    test_action( S2, Filename),
6    act(LT,S2,Filename).
```

Fonte: Autor

Na Linha (3) do Quadro 25 ocorre a adição de uma ação num registrador que representa a cadeia de ações que foram executadas no domínio. Em seguida a ação é escrita no log de execução e na Linha (5) os fluentes da teoria são testados. Por fim, na última linha ocorre a recursão do método para as demais ações previstas na lista.

Quadro 26 – Predicado *test_action/2*

```
1 test_action(S, Filename) :-
2    location(cpto, loc(X,Y), S),
3    forall(pstpstquerypairsAttribute(X,Y,AT,S), (call_time(AT, Ta),
   writeln(Ta:AT,Filename))),
4    forall(pstquerypairs(X,Y,Q,S), (Q -> call_time(Q, Ts),
   writeln(Ts:true:Q,Filename) ; call_time(Q, Ts), writeln(Ts:false:Q,Filename))).
```

Fonte: Autor

No Quadro 26 o predicado *pstpstquerypairsAttribute* é responsável pela consulta dos fluentes *depth/4*, *size/4*, *dist/4* e *dist/5* após cada ação executada, e o predicado *pstquerypairs* é responsável pela consulta dos fluentes de transições entre pares de picos *approaching/4*, *coalescing/4*, *receding/4*, *splitting/4*, *hiding/4* e *two_peak_static/4*.

Neste cenário constam dois objetos denominados $bodyA$ e $bodyB$ dentro do campo de visão do robô, posicionados respectivamente nos locais $loc(-7, -2)$ e $loc(5,1)$. A trajetória do robô se inicia no local $loc(0.0,8.0)$ e termina no mesmo local após uma volta completa em cada sentido. Durante o percurso, espera-se que o robô inicie o teste com a percepção de que o objeto $bodyA$ está à direita do objeto $bodyB$ e conforme o robô $cpto$ for se movimentando, o objeto $bodyA$ passará por trás do objeto $bodyB$ até que o lado oposto do campo de visão seja alcançado passando por todos os axiomas de transição para pares de picos.

5.4.2 Teste do fluente *approaching/4*

Esta subseção tem o objetivo de testar o fluente *approaching/4* e compreende sequências de ações que fazem com que o robô móvel descreva uma trajetória, de modo que do seu ponto de vista, 2 picos de profundidade que representam os corpos $bodyA$ e $bodyB$ se aproximam. Neste

teste a aproximação foi verificada nos 4 quadrantes possíveis, nos sentidos horário (Quadro 27 e Quadro 28) e anti-horário (Quadro 29 e Quadro 30).

Quadro 27 – Teste 1 do fluente *approaching/4*

```
1  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(0.0, 8.0),
2      do(startMove(cpto,loc(0.0,8.0),loc(1.83,7.93),20),
3          do(endMove(cpto,loc(0.1,8.1),loc(0.0,8.0),1),
4              do(startMove(cpto,loc(0.1,8.1),loc(0.0,8.0),1),
5                  do(endMove(cpto,loc(0.0,8.0),loc(0.1,8.1),1),
6                      do(startMove(cpto,loc(0.0,8.0),loc(0.1,8.1),1),s0))))).
7  true.
8  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(1.83,7.93),
9      do(startMove(cpto,loc(1.83,7.93),loc(3.62,7.73),24),
10         do(endPan(-7.5,23),do(startPan(-7.5,22),
11             do(endMove(cpto,loc(0.0,8.0),loc(1.83,7.93),21),
12                 do(startMove(cpto,loc(0.0,8.0),loc(1.83,7.93),20),
13                     do(endMove(cpto,loc(0.1,8.1),loc(0.0,8.0),1),
14                         do(startMove(cpto,loc(0.1,8.1),loc(0.0,8.0),1),
15                             do(endMove(cpto,loc(0.0,8.0),loc(0.1,8.1),1),
16                                 do(startMove(cpto,loc(0.0,8.0),loc(0.1,8.1),1),s0)))))))))).
17 true.
```

Fonte: Autor

No teste do Quadro 27 referente ao quadrante superior no sentido horário, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(0.0,8.0)$ e inicia um movimento em direção à posição $loc(1.83,7.93)$. Esta movimentação faz com que a distância angular entre os picos de profundidade diminua de 72.03 para 67.76 graus, o que configura uma aproximação entre os picos e justifica o valor verdadeiro para o fluente *approaching/4*. Em seguida, o robô continua a se movimentar para outros locais como $loc(3.62,7.73)$ e $loc(5.36,7.39)$, e sempre que o robô iniciou o movimento o fluente apresentou o valor verdadeiro.

No teste do Quadro 28 referente ao quadrante inferior no sentido horário, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(-3.62, -7.73)$ e inicia um movimento em direção à posição $loc(-5.36, -7.39)$. Esta movimentação faz com que a distância angular entre os picos de profundidade diminua de 53.55 para 50.58 graus, o que também configura uma aproximação entre os picos e justifica o valor verdadeiro para o fluente *approaching/4*. Em seguida, o robô continua a se movimentar para outros locais como $loc(-7.0, -6.93)$ e $loc(-8.52, -6.35)$, e sempre que o robô iniciou o movimento o fluente apresentou o valor verdadeiro. Devido a extensão da representação da situação corrente neste método, apenas as 5 ações que precedem o teste do fluente estão descritas no Quadro 28.

No teste do Quadro 29 referente ao quadrante superior no sentido anti-horário, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(-1.83,7.93)$ e inicia um movimento em direção à posição $loc(-3.62,7.73)$. Esta movimentação faz com que a dis-

Quadro 28 – Teste 2 do fluente *approaching/4*

```
1  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(-3.62,-7.73),
2      do(startMove(cpto,loc(-3.62,-7.73),loc(-5.36,-7.39),124),
3          do(endPan(-7.5,123),
4              do(startPan(-7.5,122),
5                  do(endMove(cpto,loc(-1.83,-7.93),loc(-3.62,-7.73),121),
6                      do(startMove(cpto,loc(-1.83,-7.93),loc(-3.62,-7.73),120),...).
7  true.
8  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(-7.0,-6.93),
9      do(startMove(cpto,loc(-7.0,-6.93),loc(-8.52,-6.35),132),
10         do(endPan(-7.5,131),
11             do(startPan(-7.5,130),
12                 do(endMove(cpto,loc(-5.36,-7.39),loc(-7.0,-6.93),129),
13                     do(startMove(cpto,loc(-5.36,-7.39),loc(-7.0,-6.93),128),...).
14 true.
```

Fonte: Autor

Quadro 29 – Teste 3 do fluente *approaching/4*

```
1  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(-1.83,7.93),
2      do(startMove(cpto,loc(-1.83,7.93),loc(-3.62,7.73),24),
3          do(endPan(7.5,23),
4              do(startPan(7.5,22),
5                  do(endMove(cpto,loc(0.0,8.0),loc(-1.83,7.93),21),
6                      do(startMove(cpto,loc(0.0,8.0),loc(-1.83,7.93),20),...).
7  true.
8  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(-5.36,7.39),
9      do(startMove(cpto,loc(-5.36,7.39),loc(-7.0,6.93),32),
10         do(endPan(7.5,31),
11             do(startPan(7.5,30),
12                 do(endMove(cpto,loc(-3.62,7.73),loc(-5.36,7.39),29),
13                     do(startMove(cpto,loc(-3.62,7.73),loc(-5.36,7.39),28),...).
14 true.
```

Fonte: Autor

tância angular entre os picos de profundidade diminua de 72.47 para 68.66 graus, o que também configura uma aproximação entre os picos e justifica o valor verdadeiro para o fluente *approaching/4*. Em seguida, o robô continua a se movimentar para outros locais como $loc(-5.36,7.39)$ e $loc(-7.0,6.93)$, e sempre que o robô iniciou o movimento o fluente apresentou o valor verdadeiro. Devido a extensão da representação da situação corrente neste método, apenas as 5 ações que precedem o teste do fluente estão descritas no Quadro 29.

Quadro 30 – Teste 4 do fluente *approaching/4*

```
1  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(0.0,-8.0),
2      do(startMove(cpto,loc(0.0,-8.0),loc(1.83,-7.93),116),
3          do(endPan(7.5,115),
4              do(startPan(7.5,114),
5                  do(endMove(cpto,loc(-1.83,-7.93),loc(0.0,-8.0),113),
6                      do(startMove(cpto,loc(-1.83,-7.93),loc(0.0,-8.0),112),...).
7  true.
8  ?- approaching(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(3.62,-7.73),
9      do(startMove(cpto,loc(3.62,-7.73),loc(5.36,-7.39),124),
10         do(endPan(7.5,123),
11             do(startPan(7.5,122),
12                 do(endMove(cpto,loc(1.83,-7.93),loc(3.62,-7.73),121),
13                     do(startMove(cpto,loc(1.83,-7.93),loc(3.62,-7.73),120),...).
14 true.
```

Fonte: Autor

No teste do Quadro 30 referente ao quadrante inferior no sentido anti-horário, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(0.0, -8.0)$ e inicia um movimento em direção à posição $loc(1.83, -7.93)$. Esta movimentação faz com que a distância angular entre os picos de profundidade diminua de 53.77 para 50.82 graus, o que também configura uma aproximação entre os picos e justifica o valor verdadeiro para o fluente *approaching/4*. Em seguida, o robô continua a se movimentar para outros locais como $loc(3.62, -7.73)$ e $loc(5.36, -7.39)$, e sempre que o robô iniciou o movimento o fluente apresentou o valor verdadeiro. Devido a extensão da representação da situação corrente neste método, apenas as 5 ações que precedem o teste do fluente estão descritas no Quadro 30.

A natureza desta teoria é determinística devido aos testes serem realizados de forma simulada. Por este motivo, os 4 testes do fluente *approaching/4* são suficientes para cobrir todas as possibilidades de cálculo no sistema para este fluente. Para os demais casos apenas um quadrante será descrito, porque os demais quadrantes são análogos e constam no código compartilhado no *Github*.

5.4.3 Teste do fluente *coalescing/4*

Esta subseção tem o objeto de testar o fluente *coalescing/4* e compreende sequências de ações que fazem com que o robô móvel descreva uma trajetória, de modo que do seu ponto de vista, 2 corpos que estão presentes no seu campo de visão se aproximam e começam a se fundir. Algumas ações realizadas no teste podem ser vistas no Quadro 31.

No teste do Quadro 31 referente ao fluente *coalescing/4*, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(9.9,5.66)$ e inicia um movimento em direção

Quadro 31 – Teste 1 fluente *coalescing/4*

```
1  ?- coalescing(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(9.9,5.66),
2      do(startMove(cpto,loc(9.9,5.66),loc(11.11,4.87),44),
3          do(endPan(-7.5,43),
4              do(startPan(-7.5,42),
5                  do(endMove(cpto,loc(8.52,6.35),loc(9.9,5.66),41),
6                      do(startMove(cpto,loc(8.52,6.35),loc(9.9,5.66),40),...).
7  true.
8  ?- coalescing(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(11.11,4.87),
9      do(startMove(cpto,loc(11.11,4.87),loc(12.12,4.0),48),
10         do(endPan(-7.5,47),
11             do(startPan(-7.5,46),
12                 do(endMove(cpto,loc(9.9,5.66),loc(11.11,4.87),45),
13                     do(startMove(cpto,loc(9.9,5.66),loc(11.11,4.87),44),...).
14 true.
```

Fonte: Autor

à posição $loc(11.11,4.87)$. Esta movimentação faz com que a distância angular entre os picos de profundidade diminua de 19.52 para 12.27 graus, que é menor do que o limiar de fusão configurado na situação inicial do sistema que vale 15 graus. Esta situação configura o início de uma transição de fusão entre picos de profundidade, ou seja os picos estão se fundindo, e justifica o valor verdadeiro para o fluente *coalescing/4*. Em seguida, o robô continua a se movimentar para outros locais como $loc(12.12,4.0)$, e o valor do fluente também se mostrou verdadeiro. Devido a extensão da representação da situação corrente neste método, apenas as 5 ações que precedem o teste do fluente estão descritas no Quadro 31.

5.4.4 Teste do fluente *hiding/4*

Esta subseção tem o objeto de testar o fluente *hiding/4* e compreende sequências de ações que fazem com que o robô móvel descreva uma trajetória, de modo que do seu ponto de vista, 2 corpos cujos picos de profundidade estão se fundindo começam a entrar numa situação de oclusão. A separação dos fluentes *coalescing/4* e *hiding/4* ocorre através de um limiar de oclusão configurado na situação inicial do sistema. Algumas ações realizadas no teste podem ser vistas no Quadro 32.

No teste do Quadro 32 referente ao fluente *hiding/4*, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(12.12,4.0)$ e inicia um movimento em direção à posição $loc(12.93,3.06)$. Esta movimentação faz com que a distância angular entre os picos de profundidade diminua de 6.46 para 1.64 graus, que é menor do que o limiar de oclusão que vale 5 graus. Esta situação configura o início de uma transição de oclusão entre picos de profundidade, ou seja os picos estão se ocluindo, e justifica o valor verdadeiro para o fluente

Quadro 32 – Teste 1 fluente *hiding/4*

```
1  ?- hiding(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(12.12,4.0),
2      do(startMove(cpto,loc(12.12,4.0),loc(12.93,3.06),52),
3          do(endPan(-7.5,51),
4              do(startPan(-7.5,50),
5                  do(endMove(cpto,loc(11.11,4.87),loc(12.12,4.0),49),
6                      do(startMove(cpto,loc(11.11,4.87),loc(12.12,4.0),48),...).
7  true.
8  ?- hiding(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(12.93,3.06),
9      do(startMove(cpto,loc(12.93,3.06),loc(13.52,2.07),56),
10         do(endPan(-7.5,55),
11             do(startPan(-7.5,54),
12                 do(endMove(cpto,loc(12.12,4.0),loc(12.93,3.06),53),
13                     do(startMove(cpto,loc(12.12,4.0),loc(12.93,3.06),52),...).
14 true.
```

Fonte: Autor

hiding/4. Em seguida, o robô continua a se movimentar para outros locais como $loc(13.52,2.07)$ e $loc(13.88,1.04)$, e o valor do fluente também se mostrou verdadeiro de forma correta. Devido a extensão da representação da situação corrente neste método, apenas as 5 ações que precedem o teste do fluente estão descritas no Quadro 32.

5.4.5 Teste do fluente *splitting/4*

Esta subseção tem o objeto de testar o fluente *splitting/4* e compreende sequências de ações que fazem com que o robô móvel descreva uma trajetória, de modo que do seu ponto de vista, um pico de profundidade presente no seu campo de visão aumenta de tamanho e começa a se separar em 2 picos. Algumas ações realizadas no teste podem ser vistas no Quadro 33.

No teste do Quadro 33 referente ao fluente *splitting/4*, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(12.93, -3.06)$ e inicia um movimento em direção à posição $loc(12.12, -4.0)$. Esta movimentação faz com que a distância angular entre os picos de profundidade aumente de 3.63 para 8.20 graus, que é maior do que o limiar de oclusão que vale 5 graus. Esta situação configura o início de uma transição de separação de um pico de profundidade em 2 picos, ou seja os picos estão se separando, e justifica o valor verdadeiro para o fluente *splitting/4*. Em seguida, o robô continua a se movimentar para outros locais como $loc(11.11, -4.87)$, e o valor do fluente também se mostrou verdadeiro de forma correta. Devido a extensão da representação da situação corrente neste método, apenas as 5 ações que precedem o teste do fluente estão descritas no Quadro 33.

Quadro 33 – Teste 1 fluente *splitting/4*

```
1  ?- splitting(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(12.93,-3.06),
2      do(startMove(cpto,loc(12.93,-3.06),loc(12.12,-4.0),80),
3          do(endPan(-7.5,79),
4              do(startPan(-7.5,78),
5                  do(endMove(cpto,loc(13.52,-2.07),loc(12.93,-3.06),77),
6                      do(startMove(cpto,loc(13.52,-2.07),loc(12.93,-3.06),76),...).
7  true.
8  ?- splitting(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(12.12,-4.0),
9      do(startMove(cpto,loc(12.12,-4.0),loc(11.11,-4.87),84),
10         do(endPan(-7.5,83),
11             do(startPan(-7.5,82),
12                 do(endMove(cpto,loc(12.93,-3.06),loc(12.12,-4.0),81),
13                     do(startMove(cpto,loc(12.93,-3.06),loc(12.12,-4.0),80),...).
14 true.
```

Fonte: Autor

5.4.6 Teste do fluente *receding/4*

Esta subseção tem o objeto de testar o fluente *receding/4* e compreende sequências de ações que fazem com que o robô móvel descreva uma trajetória, de modo que do seu ponto de vista, 2 picos de profundidade que estão se separando se afastam de modo que não exista mais uma sobreposição de corpos. Algumas ações realizadas no teste podem ser vistas no Quadro 34.

Quadro 34 – Teste 1 fluente *receding/4*

```
1  ?- receding(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(11.11,-4.87),
2      do(startMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),88),
3          do(endPan(-7.5,87),
4              do(startPan(-7.5,86),
5                  do(endMove(cpto,loc(12.12,-4.0),loc(11.11,-4.87),85),
6                      do(startMove(cpto,loc(12.12,-4.0),loc(11.11,-4.87),84),...).
7  true.
8  ?- receding(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(9.9,-5.66),
9      do(startMove(cpto,loc(9.9,-5.66),loc(8.52,-6.35),92),
10         do(endPan(-7.5,91),
11             do(startPan(-7.5,90),
12                 do(endMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),89),
13                     do(startMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),88),...).
14 true.
```

Fonte: Autor

No teste do Quadro 34 referente ao fluente *receding/4*, após as devidas inicializações do sistema, o robô inicia a trajetória na posição $loc(11.11, -4.87)$ e inicia um movimento em direção à posição $loc(9.9, -5.66)$. Esta movimentação faz com que a distância angular entre os picos de profundidade aumente de 13.34 para 19.24 graus, que é maior do que o limiar de fusão que vale 15 graus. Esta situação configura uma situação onde 2 picos estão se afastando, e justifica o valor verdadeiro para o fluente *receding/4*. Em seguida, o robô continua a se movimentar para outros locais como $loc(8.52, -6.35)$ e $loc(7.0, -6.93)$, e o valor do fluente também se mostrou verdadeiro de forma correta. Devido a extensão da representação da situação corrente neste método, apenas as 5 ações que precedem o teste do fluente estão descritas no Quadro 34.

5.4.7 Teste do fluente *two_peak_static/4*

Esta subseção tem o objeto de testar o fluente *two_peak_static/4* porém não possui uma sequência própria de ações. Isto se deve ao fato de que em todos os momentos em que o robô efetuou a ação *endMove* nos testes anteriores o fluente *two_peak_static/4* tornou-se verdadeiro. Isto se deve ao fato de que em cada uma dessas situações configurou-se um término de movimento. O Quadro 35 demonstra tal afirmação utilizando as ações do teste do fluente *receding/4*.

Quadro 35 – Teste 1 fluente *two_peak_static/4*

```
1  ?- two_peak_static(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(9.9,-5.66),
2      do(endMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),89),
3          do(startMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),88),
4              do(endPan(-7.5,87),
5                  do(startPan(-7.5,86),
6                      do(endMove(cpto,loc(12.12,-4.0),loc(11.11,-4.87),85),...).
7  true.
8  ?- two_peak_static(pk(bodyA,DepthA,SizeA,DistanceA),
     pk(bodyB,DepthB,SizeB,DistanceB), loc(8.52,-6.35),
9      do(endMove(cpto,loc(9.9,-5.66),loc(8.52,-6.35),93),
10         do(startMove(cpto,loc(9.9,-5.66),loc(8.52,-6.35),92),
11             do(endPan(-7.5,91),
12                 do(startPan(-7.5,90),
13                     do(endMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),89),...).
14 true.
```

Fonte: Autor

5.4.8 Resultado dos testes de transição para pares de picos

O conjunto de ações realizadas neste experimento e seus resultados podem ser vistos na Tabela 2. Nesta tabela estão descritas as ações exatamente como foram executadas no sistema em Prolog. Para cada ação verificou-se os valores dos atributos de tamanho angular dos objetos

$bodyA$ e $bodyB$ e distância angular entre eles. Também foi verificado para cada ação os valores resultantes dos axiomas de transição para pares de picos.

Tabela 2 – Resultados dos testes de transição para pares de picos

(continua)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(0.0,8.0),loc(1.83,7.93),20)	1	12.454	13.351	72.034	**true**	false	false	false	false	false
do(endMove(cpto,loc(0.0,8.0),loc(1.83,7.93),21)	2	10.789	15.081	67.762	false	false	false	false	false	**true**
do(startPan(-7.5,22)	3	10.789	15.081	67.762	false	false	false	false	false	**true**
do(endPan(-7.5,23)	4	10.789	15.081	67.762	false	false	false	false	false	**true**
do(startMove(cpto,loc(1.83,7.93),loc(3.62,7.73),24)	5	10.789	15.081	67.762	**true**	false	false	false	false	false
do(endMove(cpto,loc(1.83,7.93),loc(3.62,7.73),25)	6	9.507	16.739	60.116	false	false	false	false	false	**true**
do(startPan(-7.5,26)	7	9.507	16.739	60.116	false	false	false	false	false	**true**
do(endPan(-7.5,27)	8	9.507	16.739	60.116	false	false	false	false	false	**true**
do(startMove(cpto,loc(3.62,7.73),loc(5.36,7.39),28)	9	9.507	16.739	60.116	**true**	false	false	false	false	false
do(endMove(cpto,loc(3.62,7.73),loc(5.36,7.39),29)	10	8.506	17.978	49.972	false	false	false	false	false	**true**
do(startPan(-7.5,30)	11	8.506	17.978	49.972	false	false	false	false	false	**true**
do(endPan(-7.5,31)	12	8.506	17.978	49.972	false	false	false	false	false	**true**
do(startMove(cpto,loc(5.36,7.39),loc(7.0,6.93),32)	13	8.506	17.978	49.972	**true**	false	false	false	false	false
do(endMove(cpto,loc(5.36,7.39),loc(7.0,6.93),33)	14	7.726	18.389	38.905	false	false	false	false	false	**true**
do(startPan(-7.5,34)	15	7.726	18.389	38.905	false	false	false	false	false	**true**
do(endPan(-7.5,35)	16	7.726	18.389	38.905	false	false	false	false	false	**true**
do(startMove(cpto,loc(7.0,6.93),loc(8.52,6.35),36)	17	7.726	18.389	38.905	**true**	false	false	false	false	false
do(endMove(cpto,loc(7.0,6.93),loc(8.52,6.35),37)	18	7.114	17.967	28.460	false	false	false	false	false	**true**
do(startPan(-7.5,38)	19	7.114	17.967	28.460	false	false	false	false	false	**true**
do(endPan(-7.5,39)	20	7.114	17.967	28.460	false	false	false	false	false	**true**
do(startMove(cpto,loc(8.52,6.35),loc(9.9,5.66),40)	21	7.114	17.967	28.460	**true**	false	false	false	false	false
do(endMove(cpto,loc(8.52,6.35),loc(9.9,5.66),41)	22	6.631	17.009	19.523	false	false	false	false	false	**true**
do(startPan(-7.5,42)	23	6.631	17.009	19.523	false	false	false	false	false	**true**
do(endPan(-7.5,43)	24	6.631	17.009	19.523	false	false	false	false	false	**true**
do(startMove(cpto,loc(9.9,5.66),loc(11.11,4.87),44)	25	6.631	17.009	19.523	false	**true**	false	false	false	false
do(endMove(cpto,loc(9.9,5.66),loc(11.11,4.87),45)	26	6.253	15.895	12.271	false	false	false	false	false	**true**
do(startPan(-7.5,46)	27	6.253	15.895	12.271	false	false	false	false	false	**true**
do(endPan(-7.5,47)	28	6.253	15.895	12.271	false	false	false	false	false	**true**
do(startMove(cpto,loc(11.11,4.87),loc(12.12,4.0),48)	29	6.253	15.895	12.271	false	**true**	false	false	false	false
do(endMove(cpto,loc(11.11,4.87),loc(12.12,4.0),49)	30	5.963	14.873	6.458	false	false	false	false	false	**true**

Tabela 2 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startPan(-7.5,50)	31	5.963	14.873	6.458	false	false	false	false	false	**true**
do(endPan(-7.5,51)	32	5.963	14.873	6.458	false	false	false	false	false	**true**
do(startMove(cpto,loc(12.12,4.0),loc(12.93,3.06),52)	33	5.963	14.873	6.458	false	false	**true**	false	false	false
do(endMove(cpto,loc(12.12,4.0),loc(12.93,3.06),53)	34	5.744	14.021	1.635	false	false	false	false	false	**true**
do(startPan(-7.5,54)	35	5.744	14.021	1.635	false	false	false	false	false	**true**
do(endPan(-7.5,55)	36	5.744	14.021	1.635	false	false	false	false	false	**true**
do(startMove(cpto,loc(12.93,3.06),loc(13.52,2.07),56)	37	5.744	14.021	1.635	false	false	**true**	false	false	false
do(endMove(cpto,loc(12.93,3.06),loc(13.52,2.07),57)	38	5.587	13.375	-2.518	false	false	false	false	false	**true**
do(startPan(-7.5,58)	39	5.587	13.375	-2.518	false	false	false	false	false	**true**
do(endPan(-7.5,59)	40	5.587	13.375	-2.518	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.52,2.07),loc(13.88,1.04),60)	41	5.587	13.375	-2.518	false	false	**true**	false	false	false
do(endMove(cpto,loc(13.52,2.07),loc(13.88,1.04),61)	42	5.484	12.932	-6.318	false	false	false	false	false	**true**
do(startPan(-7.5,62)	43	5.484	12.932	-6.318	false	false	false	false	false	**true**
do(endPan(-7.5,63)	44	5.484	12.932	-6.318	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.88,1.04),loc(14.0,0.0),64)	45	5.484	12.932	-6.318	false	false	false	false	false	false
do(endMove(cpto,loc(13.88,1.04),loc(14.0,0.0),65)	46	5.434	12.680	-8.157	false	false	false	false	false	**true**
do(startPan(-7.5,66)	47	5.434	12.680	-8.157	false	false	false	false	false	**true**
do(endPan(-7.5,67)	48	5.434	12.680	-8.157	false	false	false	false	false	**true**
do(startMove(cpto,loc(14.0,0.0),loc(13.88,-1.04),68)	49	5.434	12.680	-8.157	false	false	false	false	false	false
do(endMove(cpto,loc(14.0,0.0),loc(13.88,-1.04),69)	50	5.433	12.602	-4.363	false	false	false	false	false	**true**
do(startPan(-7.5,70)	51	5.433	12.602	-4.363	false	false	false	false	false	**true**
do(endPan(-7.5,71)	52	5.433	12.602	-4.363	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.88,-1.04),loc(13.52,-2.07),72)	53	5.433	12.602	-4.363	false	false	false	false	false	false
do(endMove(cpto,loc(13.88,-1.04),loc(13.52,-2.07),73)	54	5.480	12.679	-.483	false	false	false	false	false	**true**
do(startPan(-7.5,74)	55	5.480	12.679	-.483	false	false	false	false	false	**true**
do(endPan(-7.5,75)	56	5.480	12.679	-.483	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.52,-2.07),loc(12.93,-3.06),76)	57	5.480	12.679	-.483	false	false	false	false	false	false
do(endMove(cpto,loc(13.52,-2.07),loc(12.93,-3.06),77)	58	5.575	12.890	3.633	false	false	false	false	false	**true**
do(startPan(-7.5,78)	59	5.575	12.890	3.633	false	false	false	false	false	**true**
do(endPan(-7.5,79)	60	5.575	12.890	3.633	false	false	false	false	false	**true**
do(startMove(cpto,loc(12.93,-3.06),loc(12.12,-4.0),80)	61	5.575	12.890	3.633	false	false	false	**true**	false	false
do(endMove(cpto,loc(12.93,-3.06),loc(12.12,-4.0),81)	62	5.721	13.200	8.196	false	false	false	false	false	**true**

Tabela 2 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startPan(-7.5,82)	63	5.721	13.200	8.196	false	false	false	false	false	**true**
do(endPan(-7.5,83)	64	5.721	13.200	8.196	false	false	false	false	false	**true**
do(startMove(cpto,loc(12.12,-4.0),loc(11.11,-4.87),84)	65	5.721	13.200	8.196	false	false	false	**true**	false	false
do(endMove(cpto,loc(12.12,-4.0),loc(11.11,-4.87),85)	66	5.919	13.556	13.341	false	false	false	false	false	**true**
do(startPan(-7.5,86)	67	5.919	13.556	13.341	false	false	false	false	false	**true**
do(endPan(-7.5,87)	68	5.919	13.556	13.341	false	false	false	false	false	**true**
do(startMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),88)	69	5.919	13.556	13.341	false	false	false	false	**true**	false
do(endMove(cpto,loc(11.11,-4.87),loc(9.9,-5.66),89)	70	6.179	13.893	19.238	false	false	false	false	false	**true**
do(startPan(-7.5,90)	71	6.179	13.893	19.238	false	false	false	false	false	**true**
do(endPan(-7.5,91)	72	6.179	13.893	19.238	false	false	false	false	false	**true**
do(startMove(cpto,loc(9.9,-5.66),loc(8.52,-6.35),92)	73	6.179	13.893	19.238	false	false	false	false	**true**	false
do(endMove(cpto,loc(9.9,-5.66),loc(8.52,-6.35),93)	74	6.506	14.097	25.828	false	false	false	false	false	**true**
do(startPan(-7.5,94)	75	6.506	14.097	25.828	false	false	false	false	false	**true**
do(endPan(-7.5,95)	76	6.506	14.097	25.828	false	false	false	false	false	**true**
do(startMove(cpto,loc(8.52,-6.35),loc(7.0,-6.93),96)	77	6.506	14.097	25.828	false	false	false	false	**true**	false
do(endMove(cpto,loc(8.52,-6.35),loc(7.0,-6.93),97)	78	6.905	14.047	32.837	false	false	false	false	false	**true**
do(startPan(-7.5,98)	79	6.905	14.047	32.837	false	false	false	false	false	**true**
do(endPan(-7.5,99)	80	6.905	14.047	32.837	false	false	false	false	false	**true**
do(startMove(cpto,loc(7.0,-6.93),loc(5.36,-7.39),100)	81	6.905	14.047	32.837	false	false	false	false	**true**	false
do(endMove(cpto,loc(7.0,-6.93),loc(5.36,-7.39),101)	82	7.388	13.678	39.786	false	false	false	false	false	**true**
do(startPan(-7.5,102)	83	7.388	13.678	39.786	false	false	false	false	false	**true**
do(endPan(-7.5,103)	84	7.388	13.678	39.786	false	false	false	false	false	**true**
do(startMove(cpto,loc(5.36,-7.39),loc(3.62,-7.73),104)	85	7.388	13.678	39.786	false	false	false	false	**true**	false
do(endMove(cpto,loc(5.36,-7.39),loc(3.62,-7.73),105)	86	7.962	12.993	46.009	false	false	false	false	false	**true**
do(startPan(-7.5,106)	87	7.962	12.993	46.009	false	false	false	false	false	**true**
do(endPan(-7.5,107)	88	7.962	12.993	46.009	false	false	false	false	false	**true**
do(startMove(cpto,loc(3.62,-7.73),loc(1.83,-7.93),108)	89	7.962	12.993	46.009	false	false	false	false	**true**	false
do(endMove(cpto,loc(3.62,-7.73),loc(1.83,-7.93),109)	90	8.632	12.115	50.815	false	false	false	false	false	**true**
do(startPan(-7.5,110)	91	8.632	12.115	50.815	false	false	false	false	false	**true**
do(endPan(-7.5,111)	92	8.632	12.115	50.815	false	false	false	false	false	**true**
do(startMove(cpto,loc(1.83,-7.93),loc(0.0,-8.0),112)	93	8.632	12.115	50.815	false	false	false	false	**true**	false
do(endMove(cpto,loc(1.83,-7.93),loc(0.0,-8.0),113)	94	9.398	11.148	53.774	false	false	false	false	false	**true**

Tabela 2 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startPan(-7.5,114)	95	9.398	11.148	53.774	false	false	false	false	false	**true**
do(endPan(-7.5,115)	96	9.398	11.148	53.774	false	false	false	false	false	**true**
do(startMove(cpto,loc(0.0,-8.0),loc(-1.83,-7.93),116)	97	9.398	11.148	53.774	false	false	false	false	**true**	false
do(endMove(cpto,loc(0.0,-8.0),loc(-1.83,-7.93),117)	98	10.249	10.206	54.686	false	false	false	false	false	**true**
do(startPan(-7.5,118)	99	10.249	10.206	54.686	false	false	false	false	false	**true**
do(endPan(-7.5,119)	100	10.249	10.206	54.686	false	false	false	false	false	**true**
do(startMove(cpto,loc(-1.83,-7.93),loc(-3.62,-7.73),120)	101	10.249	10.206	54.686	**true**	false	false	false	false	false
do(endMove(cpto,loc(-1.83,-7.93),loc(-3.62,-7.73),121)	102	11.143	9.351	53.546	false	false	false	false	false	**true**
do(startPan(-7.5,122)	103	11.143	9.351	53.546	false	false	false	false	false	**true**
do(endPan(-7.5,123)	104	11.143	9.351	53.546	false	false	false	false	false	**true**
do(startMove(cpto,loc(-3.62,-7.73),loc(-5.36,-7.39),124)	105	11.143	9.351	53.546	**true**	false	false	false	false	false
do(endMove(cpto,loc(-3.62,-7.73),loc(-5.36,-7.39),125)	106	12.044	8.604	50.581	false	false	false	false	false	**true**
do(startPan(-7.5,126)	107	12.044	8.604	50.581	false	false	false	false	false	**true**
do(endPan(-7.5,127)	108	12.044	8.604	50.581	false	false	false	false	false	**true**
do(startMove(cpto,loc(-5.36,-7.39),loc(-7.0,-6.93),128)	109	12.044	8.604	50.581	**true**	false	false	false	false	false
do(endMove(cpto,loc(-5.36,-7.39),loc(-7.0,-6.93),129)	110	12.859	7.973	46.126	false	false	false	false	false	**true**
do(startPan(-7.5,130)	111	12.859	7.973	46.126	false	false	false	false	false	**true**
do(endPan(-7.5,131)	112	12.859	7.973	46.126	false	false	false	false	false	**true**
do(startMove(cpto,loc(-7.0,-6.93),loc(-8.52,-6.35),132)	113	12.859	7.973	46.126	**true**	false	false	false	false	false
do(endMove(cpto,loc(-7.0,-6.93),loc(-8.52,-6.35),133)	114	13.533	7.452	40.660	false	false	false	false	false	**true**
do(startPan(-7.5,134)	115	13.533	7.452	40.660	false	false	false	false	false	**true**
do(endPan(-7.5,135)	116	13.533	7.452	40.660	false	false	false	false	false	**true**
do(startMove(cpto,loc(-8.52,-6.35),loc(-9.9,-5.66),136)	117	13.533	7.452	40.660	**true**	false	false	false	false	false
do(endMove(cpto,loc(-8.52,-6.35),loc(-9.9,-5.66),137)	118	14.026	7.026	34.655	false	false	false	false	false	**true**
do(startPan(-7.5,138)	119	14.026	7.026	34.655	false	false	false	false	false	**true**
do(endPan(-7.5,139)	120	14.026	7.026	34.655	false	false	false	false	false	**true**
do(startMove(cpto,loc(-9.9,-5.66),loc(-11.11,-4.87),140)	121	14.026	7.026	34.655	**true**	false	false	false	false	false
do(endMove(cpto,loc(-9.9,-5.66),loc(-11.11,-4.87),141)	122	14.351	6.687	28.570	false	false	false	false	false	**true**
do(startPan(-7.5,142)	123	14.351	6.687	28.570	false	false	false	false	false	**true**
do(endPan(-7.5,143)	124	14.351	6.687	28.570	false	false	false	false	false	**true**
do(startMove(cpto,loc(-11.11,-4.87),loc(-12.12,-4.0),144)	125	14.351	6.687	28.570	**true**	false	false	false	false	false
do(endMove(cpto,loc(-11.11,-4.87),loc(-12.12,-4.0),145)	126	14.567	6.428	22.746	false	false	false	false	false	**true**

Tabela 2 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startPan(-7.5,146)	127	14.567	6.428	22.746	false	false	false	false	false	**true**
do(endPan(-7.5,147)	128	14.567	6.428	22.746	false	false	false	false	false	**true**
do(startMove(cpto,loc(-12.12,-4.0),loc(-12.93,-3.06),148)	129	14.567	6.428	22.746	**true**	false	false	false	false	false
do(endMove(cpto,loc(-12.12,-4.0),loc(-12.93,-3.06),149)	130	14.740	6.236	17.227	false	false	false	false	false	**true**
do(startPan(-7.5,150)	131	14.740	6.236	17.227	false	false	false	false	false	**true**
do(endPan(-7.5,151)	132	14.740	6.236	17.227	false	false	false	false	false	**true**
do(startMove(cpto,loc(-12.93,-3.06),loc(-13.52,-2.07),152)	133	14.740	6.236	17.227	false	**true**	false	false	false	false
do(endMove(cpto,loc(-12.93,-3.06),loc(-13.52,-2.07),153)	134	14.951	6.107	12.032	false	false	false	false	false	**true**
do(startPan(-7.5,154)	135	14.951	6.107	12.032	false	false	false	false	false	**true**
do(endPan(-7.5,155)	136	14.951	6.107	12.032	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.52,-2.07),loc(-13.88,-1.04),156)	137	14.951	6.107	12.032	false	**true**	false	false	false	false
do(endMove(cpto,loc(-13.52,-2.07),loc(-13.88,-1.04),157)	138	15.280	6.037	7.013	false	false	false	false	false	**true**
do(startPan(-7.5,158)	139	15.280	6.037	7.013	false	false	false	false	false	**true**
do(endPan(-7.5,159)	140	15.280	6.037	7.013	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.88,-1.04),loc(-14.0,-0.0),160)	141	15.280	6.037	7.013	false	false	**true**	false	false	false
do(endMove(cpto,loc(-13.88,-1.04),loc(-14.0,-0.0),161)	142	15.790	6.026	2.025	false	false	false	false	false	**true**
do(startPan(-7.5,162)	143	15.790	6.026	2.025	false	false	false	false	false	**true**
do(endPan(-7.5,163)	144	15.790	6.026	2.025	false	false	false	false	false	**true**
do(startMove(cpto,loc(-14.0,-0.0),loc(-13.88,1.04),164)	145	15.790	6.026	2.025	false	false	**true**	false	false	false
do(endMove(cpto,loc(-14.0,-0.0),loc(-13.88,1.04),165)	146	16.553	6.072	-3.248	false	false	false	false	false	**true**
do(startPan(-7.5,166)	147	16.553	6.072	-3.248	false	false	false	false	false	**true**
do(endPan(-7.5,167)	148	16.553	6.072	-3.248	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.88,1.04),loc(-13.52,2.07),168)	149	16.553	6.072	-3.248	false	false	**true**	false	false	false
do(endMove(cpto,loc(-13.88,1.04),loc(-13.52,2.07),169)	150	17.644	6.180	-9.221	false	false	false	false	false	**true**
do(startPan(-7.5,170)	151	17.644	6.180	-9.221	false	false	false	false	false	**true**
do(endPan(-7.5,171)	152	17.644	6.180	-9.221	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.52,2.07),loc(-12.93,3.06),172)	153	17.644	6.180	-9.221	false	false	false	false	false	false
do(endMove(cpto,loc(-13.52,2.07),loc(-12.93,3.06),173)	154	19.111	6.353	-9.151	false	false	false	false	false	**true**
do(startPan(-7.5,174)	155	19.111	6.353	-9.151	false	false	false	false	false	**true**
do(endPan(-7.5,175)	156	19.111	6.353	-9.151	false	false	false	false	false	**true**
do(startMove(cpto,loc(-12.93,3.06),loc(-12.12,4.0),176)	157	19.111	6.353	-9.151	false	false	false	false	false	false
do(endMove(cpto,loc(-12.93,3.06),loc(-12.12,4.0),177)	158	20.964	6.597	-2.383	false	false	false	false	false	**true**

Tabela 2 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startPan(-7.5,178)	159	20.964	6.597	-2.383	false	false	false	false	false	**true**
do(endPan(-7.5,179)	160	20.964	6.597	-2.383	false	false	false	false	false	**true**
do(startMove(cpto,loc(-12.12,4.0),loc(-11.11,4.87),180)	161	20.964	6.597	-2.383	false	false	false	**true**	false	false
do(endMove(cpto,loc(-12.12,4.0),loc(-11.11,4.87),181)	162	23.014	6.921	6.451	false	false	false	false	false	**true**
do(startPan(-7.5,182)	163	23.014	6.921	6.451	false	false	false	false	false	**true**
do(endPan(-7.5,183)	164	23.014	6.921	6.451	false	false	false	false	false	**true**
do(startMove(cpto,loc(-11.11,4.87),loc(-9.9,5.66),184)	165	23.014	6.921	6.451	false	false	false	false	**true**	false
do(endMove(cpto,loc(-11.11,4.87),loc(-9.9,5.66),185)	166	24.731	7.345	18.203	false	false	false	false	false	**true**
do(startPan(-7.5,186)	167	24.731	7.345	18.203	false	false	false	false	false	**true**
do(endPan(-7.5,187)	168	24.731	7.345	18.203	false	false	false	false	false	**true**
do(startMove(cpto,loc(-9.9,5.66),loc(-8.52,6.35),188)	169	24.731	7.345	18.203	false	false	false	false	**true**	false
do(endMove(cpto,loc(-9.9,5.66),loc(-8.52,6.35),189)	170	25.068	7.887	32.672	false	false	false	false	false	**true**
do(startPan(-7.5,190)	171	25.068	7.887	32.672	false	false	false	false	false	**true**
do(endPan(-7.5,191)	172	25.068	7.887	32.672	false	false	false	false	false	**true**
do(startMove(cpto,loc(-8.52,6.35),loc(-7.0,6.93),192)	173	25.068	7.887	32.672	false	false	false	false	**true**	false
do(endMove(cpto,loc(-8.52,6.35),loc(-7.0,6.93),193)	174	23.406	8.569	47.715	false	false	false	false	false	**true**
do(startPan(-7.5,194)	175	23.406	8.569	47.715	false	false	false	false	false	**true**
do(endPan(-7.5,195)	176	23.406	8.569	47.715	false	false	false	false	false	**true**
do(startMove(cpto,loc(-7.0,6.93),loc(-5.36,7.39),196)	177	23.406	8.569	47.715	false	false	false	false	**true**	false
do(endMove(cpto,loc(-7.0,6.93),loc(-5.36,7.39),197)	178	20.448	9.425	60.321	false	false	false	false	false	**true**
do(startPan(-7.5,198)	179	20.448	9.425	60.321	false	false	false	false	false	**true**
do(endPan(-7.5,199)	180	20.448	9.425	60.321	false	false	false	false	false	**true**
do(startMove(cpto,loc(-5.36,7.39),loc(-3.62,7.73),200)	181	20.448	9.425	60.321	false	false	false	false	**true**	false
do(endMove(cpto,loc(-5.36,7.39),loc(-3.62,7.73),201)	182	17.291	10.493	68.663	false	false	false	false	false	**true**
do(startPan(-7.5,202)	183	17.291	10.493	68.663	false	false	false	false	false	**true**
do(endPan(-7.5,203)	184	17.291	10.493	68.663	false	false	false	false	false	**true**
do(startMove(cpto,loc(-3.62,7.73),loc(-1.83,7.93),204)	185	17.291	10.493	68.663	false	false	false	false	**true**	false
do(endMove(cpto,loc(-3.62,7.73),loc(-1.83,7.93),205)	186	14.605	11.798	72.465	false	false	false	false	false	**true**
do(startPan(-7.5,206)	187	14.605	11.798	72.465	false	false	false	false	false	**true**
do(endPan(-7.5,207)	188	14.605	11.798	72.465	false	false	false	false	false	**true**
do(startMove(cpto,loc(-1.83,7.93),loc(-0.0,8.0),208)	189	14.605	11.798	72.465	**true**	false	false	false	false	false
do(endMove(cpto,loc(-1.83,7.93),loc(-0.0,8.0),209)	190	12.454	13.351	72.034	false	false	false	false	false	**true**

Tabela 2 – Resultados dos testes de transição para pares de picos

(conclusão)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startPan(-7.5,210)	191	12.454	13.351	72.034	false	false	false	false	false	**true**
do(endPan(-7.5,211)	192	12.454	13.351	72.034	false	false	false	false	false	**true**

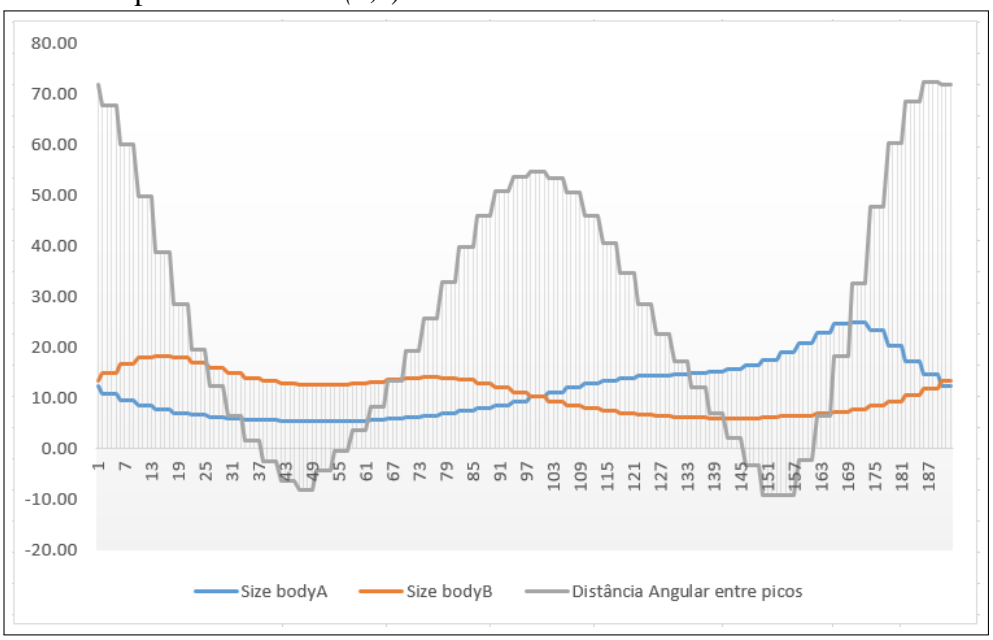
Fonte: Autor

Neste teste o robô efetua um percurso circular de 360 graus ao redor dos objeto $bodyA$ e $bodyB$ enquanto observa o centro da trajetória. Durante o teste os picos de profundidade em seu campo de visão iniciam o teste separados. Os picos se aproximam ao longo do tempo até se fundirem completamente. Em seguida se separam e se afastam. Este processo ocorre duas vezes, sendo a primeira no quadrante superior da trajetória e a segunda no quadrante inferior.

Durante a execução das ações 45, 49, 53, 57, 153 e 157 nenhum fluente de transição entre pares de picos foi verdadeiro. Isto se deve ao fato de que os corpos $bodyA$ e $bodyB$ estavam em completo estado de oclusão e por este motivo entende-se de que existe apenas um pico de profundidade no perfil gerado pelos sensores de uma robô móvel.

Na Figura (14) pode-se observar a flutuação dos valores referentes ao tamanho angular dos picos de profundidade que representam os corpos $bodyA$ e $bodyB$ observados, e também da distância angular entre os picos. Devido à trajetória de testes ser circular, conforme o robô

Figura 14 – Flutuação de tamanho e distância angular dos picos de profundidade *do(a,s)*



Fonte: Autor

móvel se aproximou de uma extremidade esquerda ou direita, o objeto que se encontrava mais próximo se mostrou aparentemente maior do que o objeto que estava mais longe. Interessante notar que os vales formados pelos valores de distância angular entre os picos representam os momentos em que houve a oclusão entre os corpos, e são os pontos onde a percepção do robô em relação aos corpos apresenta a maior disparidade em tamanho. De forma inversa, os picos de distância angular representam os momentos em que o robô esteve praticamente equidistante dos corpos, e são os pontos onde o robô possui a percepção de tamanho dos objetos com a menor disparidade.

A Tabela (2) refere-se apenas ao sentido horário dos testes. No Apêndice B consta a tabela referente ao sentido anti-horário.

6 RESULTADOS E DISCUSSÃO

Durante os testes realizados foi observado que, cada vez que uma ação do tipo *startMove/4* ou *startPan/2* acontece, um fluente de transição e somente um fluente torna-se verdadeiro. Da mesma forma que, sempre que uma ação do tipo *endMove/4* ou *endPan/2* ocorre, o fluente de transição torna-se falso e os picos permanecem estáticos. Esses testes demonstraram que, em ambiente bastante controlado, a teoria proposta em Soutchanski e Santos (2008) é correta e completa em respeito a movimento. Isto é, para cada situação gerada após a execução de ações no domínio, somente um fluente de transição foi verdadeiro e, não houve uma situação em que não existisse uma relação que explicasse a situação dos objetos, quando dentro do campo de visão do robô. Isso foi provado analiticamente em Soutchanski e Santos (2008) mas nunca antes verificado na prática. Esta dissertação descreve a primeira implementação do trabalho de Soutchanski e Santos (2008).

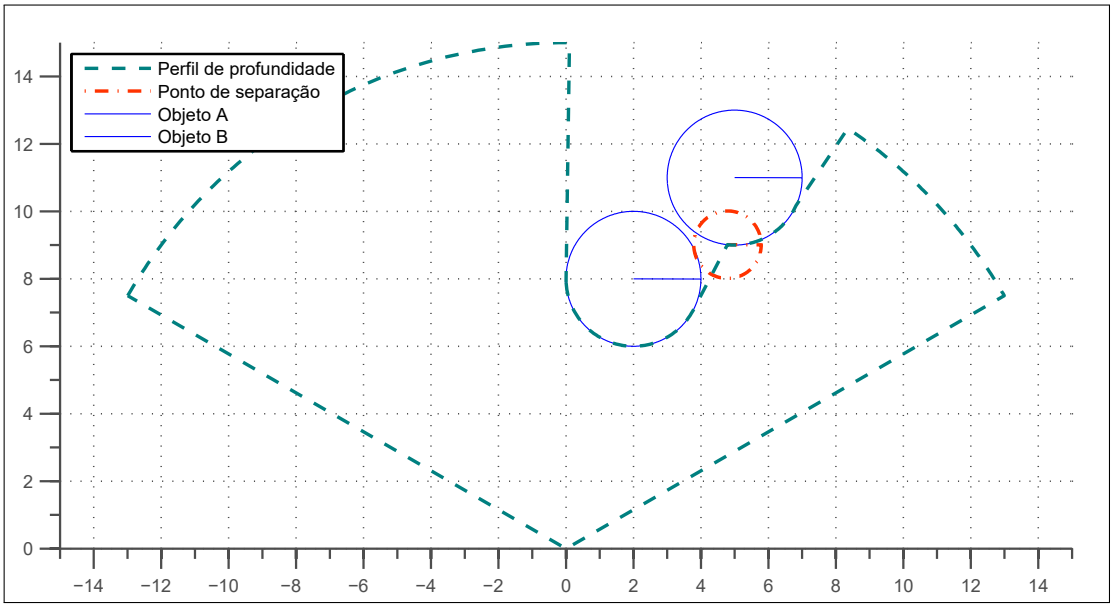
Imagine uma situação em que o robô não soubesse da existência de um dos objetos ($bodyA$), e a situação corrente configurasse um estado de oclusão, pois o objeto se aproximou, por exemplo, diretamente pelo ponto de oclusão gerado pelo objeto $bodyB$, e que até então não havia sido percebido. Caso as mesmas ações 61, 65 e 161 fossem executadas e os picos de profundidade gerados fossem os mesmos, a teoria seria incapaz de inferir o fluente *splitting/4* como verdadeiro apenas se baseando no raciocínio sobre as ações de movimento do robô, pois não há informação armazenada de que existe um segundo objeto no campo de visão. A única possibilidade seria o sistema de sensoriamento do robô gerar um perfil de profundidade que pudesse ser analisado e constatado que se tratam de dois picos distintos. O tratamento do sinal gerado pelos sensores do robô não faz parte do escopo da teoria e também não é parte deste projeto, mas é necessário que seja comentado porque a responsabilidade de gerar a informação que levará a teoria a inferir a percepção correta nesta situação seria do sistema de visão do robô. A informação esperada pela teoria neste caso é uma lista de picos com a forma

$$pk(b, u, z, d),$$

onde b representa um corpo no perfil de profundidade, u é a disparidade de profundidade, z é o tamanho angular e d é a distância angular até a borda esquerda do campo de visão do robô. Para tanto, o sistema de visão deverá ser capaz de identificar que um perfil de profundidade possui dois picos sobrepostos (Figura 15) na iminência de separação, com certo deslocamento, e assim gerar a informação correta para o bloco lógico da teoria. Caso contrário, o único fluente que poderia ser verdadeiro seria *extending/3*.

Os axiomas de estado sucessor dos fluentes da teoria possuem cálculos sobre ações de movimento realizadas pelos objetos em cena, o que subentende-se que existe algum meio pelo qual o robô adquire conhecimento que não sejam os sensores do seu sistema de visão computacional. Se o robô não gera a informação de que foram realizadas ações no domínio por outros corpos, então este conhecimento deve ser informado por uma fonte externa. A origem desta informação poderia ser tanto um observador do ambiente em que o robô se encontra,

Figura 15 – Perfil de profundidade com 2 picos se separandov



Fonte: Autor

quanto os próprios corpos que tomaram alguma ação, sendo a segunda opção menos provável pois seria necessário que o corpo fosse um agente com inteligência similar ao robô. De toda forma, seria necessário que o robô possuísse algum protocolo de comunicação que o permita adquirir conhecimento externo.

Partindo deste princípio, de que o robô adquire conhecimento externo, os axiomas de estado sucessor que realizam cálculo sobre as ações de outros corpos não precisariam necessariamente se limitar aos limites impostos pelo alcance dos seus sensores, pois matematicamente são cálculos possíveis. É claro que tudo dependeria do objetivo do robô. Pensando na proposta original de Soutchanski e Santos (2008), que é de inferir relações espaciais qualitativas do ponto de vista egocêntrico do robô, então o conhecimento externo serviria (por exemplo) para o robô adquirir o conhecimento de que existe um objeto em estado de oclusão, posicionado atrás de um outro objeto que ele consegue observar. Assim, caso o robô execute alguma ação de movimento, que matematicamente exponha este objeto inicialmente ocluso para o ponto de vista do robô, ou seja, finalize o estado de oclusão de modo que o robô consiga observá-lo diretamente, a teoria permitiria que o robô inferisse corretamente o fluente *splitting/4*, e não apenas o fluente *extending/4*.

Porém, se o objetivo fosse apenas gerar conhecimento geral sobre o domínio em que o robô se encontra, sem a necessidade de inferir relações do ponto de vista do robô, com algumas modificações quanto à restrições que são intrínsecas ao robô na teoria atual seria possível gerar relações espaciais para outros pontos de vista. Isto é devido aos axiomas atuais já receberem como parâmetro o local onde o robô se encontra, ou seja, seu ponto de vista. No entanto, esta investigação seria motivação para um trabalho futuro.

Em relação ao teste de transições para pico individual, é interessante observar que o fluente *peak_static/3* continuou verdadeiro após o término de movimento do robô, mesmo quando

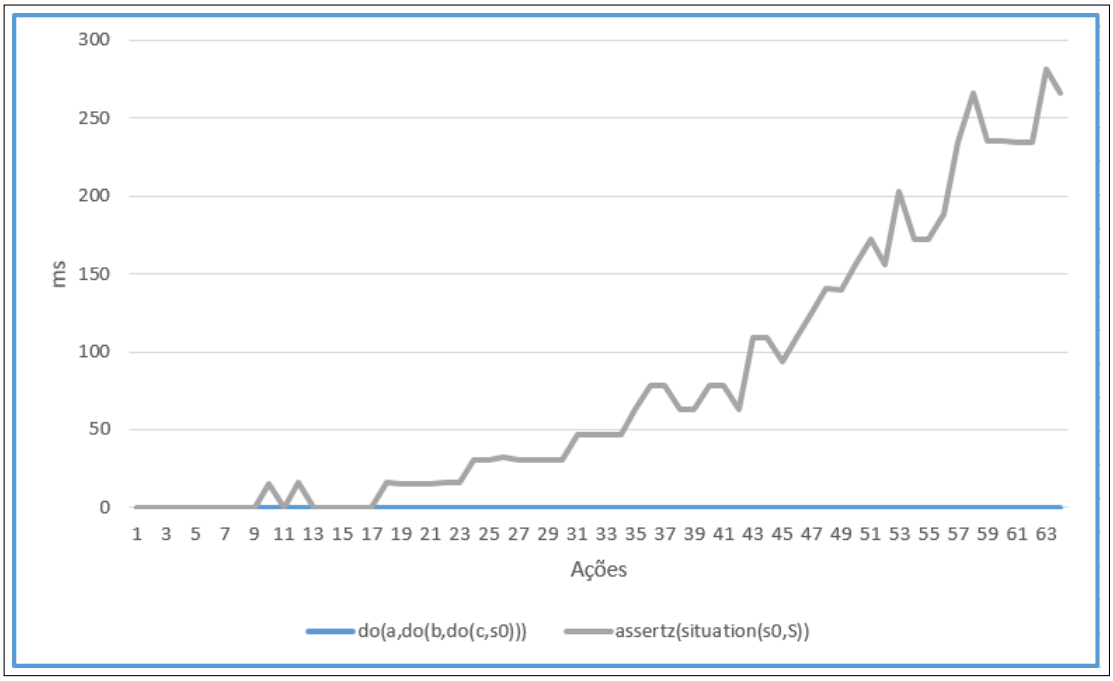
o objeto saiu do campo de visão do robô. Isto se deve ao fato de não existir nenhuma informação que prove que o objeto tenha se movimentado. Devido aos sensores do robô não possuírem alcance suficiente para que pudessem ser gerados novos perfis de profundidade e então continuasse a observar o objeto, a informação que é mantida em memória é a última localização válida do objeto em questão. A informação permanece até que o robô adquira conhecimento que altere a localização do objeto na sua percepção. É o mesmo princípio da observação efetuada em relação ao limite máximo de observação definido pelo alcance dos sensores e do limiar de profundidade. Se o robô possuir conhecimento da localização de dois objetos, que estão em seu campo de visão, e o robô efetuar continuamente um movimento que o afaste destes objetos, matematicamente seria possível continuar recalculando os valores verdade das relações *shrinking/3* e *approaching/4*. O único risco é a descaracterização dos axiomas *appearing/3* e *vanishing/3*, pois não existiria para estes casos um valor de referência para que se pudesse efetuar a separação entre *extending/4* e *shrinking/4* para *appearing/3* e *vanishing/4*.

Numa outra vertente, as ações primitivas de *startMove/4* e *endMove/4* são totalmente independentes das ações *startPan/2* e *endPan/2*, e possuem relação direta com informações diferentes sobre o robô, que respectivamente são a sua localização $loc(x_r,y_r)$ e o ângulo θ em que ele está orientado em relação ao Norte. Dessa forma, seria possível que o fluente *facing/3* não tivesse como um de seus parâmetros a informação $loc(x,y)$ representando o ponto de vista do robô. Não existe nesta teoria alguma ação de movimento que exerça influência sobre θ, apenas ações de rotação. Uma consequência do relaxamento deste parâmetro na implementação atual seria a não obrigatoriedade de se atualizar através da função *actionConsequences* o fluente *facing/3* que está armazenado em memória. Através dos axiomas de estado sucessor já seria possível obter o valor corrente de θ, assim como foi feito no segundo método do Capítulo 4.

Em relação à implementação realizada no presente trabalho, vale observar que algumas mudanças estruturais foram realizadas com o objetivo de acelerar o processamento da informação adquirida sobre o mundo. Conforme descrito no Capítulo 4, o predicado *do(A,S)* foi modificado para atender às funções *assertz/1* e *retract/1*. Estas funções adicionam e retiram dados da base de conhecimento que o Prolog usa para consultas, e por este motivo prestam manutenção da informação armazenada em memória à respeito da situação $s0$ representada pelo predicado *situation(s0, S)*. O resultado destas modificações permitiu que os cálculos de situação fossem realizados da forma esperada e prevista em Soutchanski e Santos (2008), porém não obtiveram sucesso quanto à melhoria de velocidade de processamento, tornando-se até mais custoso. Na Figura (16) pode-se observar que com apenas algumas ações o método desenvolvido que armazena a situação corrente através do functor *assertz/1* torna-se bastante lento em relação ao outro método.

Uma otimização que poderia ser realizada seria a utilização de uma lista temporária, que o sistema utilizaria para efetuar o cálculo de situação, e uma lista final onde seriam armazenadas todas as ações da situação por motivos de histórico e caracterização de situação de acordo com a definição em (REITER, 2001).

Figura 16 – Comparativo de performance entre métodos de cálculo



Fonte: Autor

Foi observado que os axiomas de estado sucessor necessitam de algumas poucas ações para serem verdadeiros. Exemplificando, isso significa que se uma ação *endMove/4* foi executada no mundo, e deseja-se testar o fluente *location/4*, basta que se observe a lista de ações temporárias até que se encontre uma ação relevante. As ações anteriores que não tivessem mais relevância para nenhum fluente seriam migradas para a lista final com o intuito de diminuir o espaço de busca para os fluentes que forem falsos.

Com esta alteração na lista de ações, a teoria teria menos informação para processar, sem que existisse a perda de dados para os cálculos. O mapeamento dos detalhes e restrições desta ideia é motivação para trabalho futuro.

7 CONCLUSÃO

Neste trabalho os axiomas descritos em Soutchanski e Santos (2008) foram implementados e testados em diversas situações que puderam fornecer dados para análise de seu funcionamento e observar questões importantes que poderão ser estudadas em trabalhos futuros.

No Capítulo 3 foi realizada uma revisão detalhada da teoria $\mathcal{TDM}$. Todas as relações espaciais e funções auxiliares foram descritas. Os axiomas que não estavam explicitamente declarados no artigo de Soutchanski e Santos (2008) por terem sido definidos por analogia, foram escritos por completo e estão presentes no Apêndice A.

No Capítulo 4 uma descrição de uma implementação em Prolog foi apresentada para que pudessem ser feitos testes automatizados. Alguns ajustes foram realizados na estrutura gerada pelo Cálculo de Situações, como medida de melhoria para que o sistema se tornasse condicionalmente menos custoso. O código possui espaço para melhoria, como a divisão da lista de ações de uma situação em duas listas separadas. Uma lista conteria dados recentes utilizados para o processamento corrente do robô móvel, e uma segunda lista armazenaria ações passadas, que não interferem mais no cálculo recente devido à restrições já existentes por outras ações do mundo, mas que continuam armazenadas para consulta do histórico.

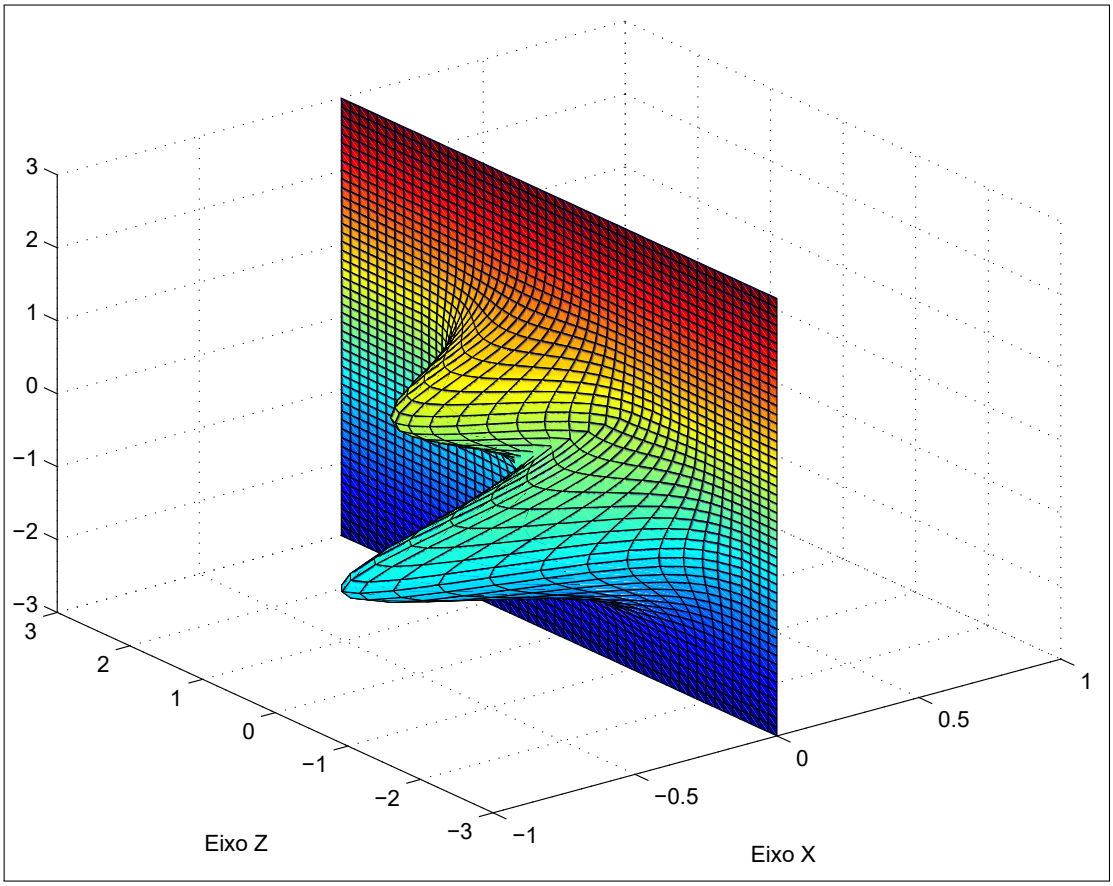
Durante o desenvolvimento do código foram realizados apontamentos sobre a formalização dos axiomas de transição, que por terem sua natureza relacionada ao movimento, não poderiam ser calculados com base na ação *endMove/4*, mas sim com base na ação *startMove/4*. Foi feita também a reformulação do axioma da relação *hiding/4*, para se diferenciar das demais transições por ser possível tanto quando há movimento, como quando todos os objetos (incluindo o observador) estão estáticos. Estes apontamentos juntamente com a definição explícita de todos os axiomas da teoria, a verificação experimental da teoria e a publicação do código desenvolvido em *https://github.com/pava197/qsrTdm-FEI* são as principais contribuições deste trabalho.

Os testes realizados foram divididos em blocos, integrantes de uma mesma composição de cena. Cada bloco consistiu de uma sequência de ações que visaram testar aspectos diferentes da teoria. O primeiro bloco testou o impacto de ações no mundo em relação à percepção espacial do robô sobre o tamanho, profundidade e distância de objetos em seu campo de visão. O segundo bloco testou as relações espaciais propostas para transições possíveis em picos de profundidade individuais. Por fim, o terceiro bloco testou as relações espaciais propostas para transições possíveis entre pares de picos profundidade.

Os resultados obtidos mostraram em ambiente controlado que a teoria responde bem às ações que acontecem no mundo e todas as relações espaciais propostas foram observadas.

O próximo passo na investigação desta teoria seria uma implementação real, com um sistema de visão computacional que gerasse os perfis de profundidade para o robô. Esta implementação poderia ser através de uma câmera estéreo ou laser, e possivelmente utilizando a tecnologia *ROS* (QUIGLEY et al., 2009). Esse passo completaria um ciclo de teste da teoria e

Figura 17 – Perfil de profundidade em 3 dimensões



Fonte: Autor

que cobriria todas as possibilidades lógicas da teoria, bem como poderia expor novos detalhes ainda não conhecidos e que precisariam ser individualmente estudados. Uma implementação com dados reais impõe diversos desafios que necessitariam ser ultrapassados para obter sucesso no experimento como a normalização dos dados observados, filtragem de ruído, identificação de separação de picos e como lidar com a contradição entre informação sensorial e resultados de ações..

Outra possibilidade que daria sequência à linha de pesquisa deste trabalho seria a expansão da teoria de Soutchanski e Santos (2008) para raciocinar em três dimensões e não apenas em duas. Na Figura 17 pode-se visualizar um exemplo de perfil de profundidade nestas condições. Considerando o sistema de localização adotado até este instante, a função que determina a localização dos objetos $loc(x_a,y_a)$ e o predicado *location/3* poderiam ser modificados para trabalhar com três parâmetros, passando a ser representados como $loc(x_a,y_a,z_a)$, e $location(Obj, loc(x_a,y_a,z_a), S)$ onde z_a é uma disparidade de altura em relação ao robô em que o objeto se encontra.

Em Santos (2007) foi proposto um corte horizontal nos dados dos sensores do robô, à meia altura, para que fossem gerados perfis de profundidade dentro desta condição. Se a teoria fosse expandida para três dimensões, e caso fossem detectados mais de um pico de profundidade no perfil, os dados obtidos pela sistema de sensoriamento do robô poderiam ser utilizados

também em forma de cortes, mas de tal forma que sempre existissem dois picos de profundidade no corte. Isto geraria informação similar à utilizada neste trabalho, com a exceção de que existiria um parâmetro adicional referente ao ângulo em que o corte foi realizado. Por exemplo, se existirem três picos p, q e r, seriam aproveitados os cortes pq, pr e qr. A referência angular serviria para determinar relações espaciais do tipo *em cima de* ou *embaixo de*. Para o caso de mais de dois objetos em cena, o raciocínio deverá ser feito aos pares.

REFERÊNCIAS

ALBATH, Julia et al. RCC-3D: qualitative spatial reasoning in 3D. Em: INTERNATIONAL CONFERENCE ON COMPUTER APPLICATIONS IN INDUSTRY AND ENGINEERING, 23., 2010, Las Vegas. **INTERNATIONAL...** Las Vegas: [s.n.], 2010. pp. 74–79. Disponível em: <https://goo.gl/EsSX9R>. Acesso em: 19 jan. 2017.

ALBATH, Julia; LEOPOLD, Jennifer L.; SABHARWAL, Chaman L. Visualization of spatio-temporal reasoning over 3D images. Em: INTERNATIONAL WORKSHOP ON VISUAL LANGUAGES AND COMPUTING, 16., 2010, Illinois. **PROCEEDINGS...** Illinois: [s.n.], 2010. vol. 10, pp. 277–282.

BHATT, Mehul; RAHAYU, J. Wenny; STERLING, Gerald. Qualitative spatial reasoning with topological relations in the situation calculus. Em: INTERNATIONAL FLORIDA ARTIFICIAL INTELLIGENCE RESEARCH SOCIETY CONFERENCE, 19., 2006, Melbourne Beach. **PROCEEDINGS...** Melbourne Beach: [s.n.], 2006. pp. 713–718. Disponível em: <https://goo.gl/G4mSNP>. Acesso em: 19 jan. 2017.

CABALAR, Pedro; SANTOS, Paulo. Formalising the fisherman's folly puzzle. **Artificial Intelligence**, Elsevier, [S.l.], vol. 175, n.º 1, pp. 346–377, 2011. Disponível em: <https://goo.gl/qdLmym>. Acesso em: 19 jan. 2017.

CHEN, Zhichao; BIRCHFIELD, Stanley T. Qualitative vision-based mobile robot navigation. Em: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2006, Hilton. **PROCEEDINGS...** Hilton: IEEE, 2006. pp. 2686–2692.

CHEVRIAUX, Yann; SAUX, Eric; CLARAMUNT, Christophe. A landform-based approach for the representation of terrain silhouettes. Em: ACM INTERNATIONAL WORKSHOP ON GEOGRAPHIC INFORMATION SYSTEMS, 13., 2005, [S.l.] **PROCEEDINGS...** [S.l.]: ACM, 2005. pp. 260–266.

COHN, Anthony G.; RENZ, Jochen. Qualitative spatial representation and reasoning. **Foundations of Artificial Intelligence**, Elsevier, [S.l.], vol. 3, pp. 551–596, 2008.

EGENHOFER, Max J.; MARK, David M. Naive geography. Em: INTERNATIONAL CONFERENCE ON SPATIAL INFORMATION THEORY, 1995, [S.l.]. **INTERNATIONAL...** [S.l.]: Springer, 1995. pp. 1–15.

FERNYHOUGH, J.; COHN, Anthony G.; HOGG, David C. Constructing qualitative event models automatically from video input. **Image and Vision Computing**, Elsevier, [S.l.], vol. 18, n.º 2, pp. 81–103, 2000.

GÄRDENFORS, Peter. **Conceptual spaces: the geometry of thought**. 1. ed. [S.l.]: MIT press, 2004.

GROSSKREUTZ, Henrik; LAKEMEYER, Gerhard. On-line execution of cc-golog plans. Em: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 17., 2001, Seattle. **INTERNATIONAL...** Seattle: [s.n.], 2001. pp. 12–18. Disponível em: <https://goo.gl/Jmh6OU>. Acesso em: 19 jan. 2017.

KREUTZMANN, Arne et. al. Towards safe navigation by formalizing navigation rules. **International Journal on Marine Navigation and Safety of Sea Transportation**, [S.l.], vol. 7, n.º 2, pp. 161–168, 2013. Disponível em: <https://goo.gl/qFPqs0>. Acesso em: 19 jan. 2017.

KUIPERS, Benjamin; BYUN, Yung-Tai. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. **Robotics and autonomous systems**, Elsevier, [S.l.], vol. 8, n.º 1, pp. 47–63, 1991.

LEVESQUE, Hector; LAKEMEYER, Gerhard. Cognitive robotics. **Foundations of artificial intelligence**, Elsevier, [S.l.], vol. 3, pp. 869–886, 2008.

LIGOZAT, Gerard; SANTOS, Paulo. Spatial occlusion and multiple views within an interval algebra. Em: AAAI SYMPOSIUM ON LOGIC FORMALIZATIONS OF COMMONSENSE REASONING., 2015, Palo Alto. **PROCEEDINGS...** Palo Alto: IEEE, 2015. Disponível em: <https://goo.gl/UJajFe>. Acesso em: 19 jan. 2017.

MICHALSKI, Ryszard S.; CARBONELL, Jaime G.; MITCHELL, Tom M. **Machine learning: an arftificial intelligence approach**. 1. ed. [S.l.]: Springer, 2013.

MORATZ, Reinhard; RENZ, Jochen; WOLTER, Diedrich. Qualitative spatial reasoning about line segments. Em: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14., 2000, Berlim. **PROCEEDINGS...** Berlim: [s.n.], 2000. pp. 234–238.

NATH, Chayanika D.; HAZARIKA, Shyamanta M. Combining diagrammatic reasoning with qualitative spatial and temporal reasoning for motion event detection. Em: NATIONAL CONFERENCE ON COMPUTER VISION, PATTERN RECOGNITION, IMAGE PROCESSING e GRAPHICS, 5., 2015, Patna. **PROCEEDINGS...** Patna: IEEE, 2015. pp. 1–4.

PEARCE, David; VALVERDE, Agustin. **Quantified equilibrium logic and the first order logic of here-and-there**. Málaga, 2006. Disponível em: <https://goo.gl/x85w5k>. Acesso em: 19 jan. 2017.

QUIGLEY, Morgan et al. ROS: an open-source robot operating system. Em: ICRA WORKSHOP ON OPEN SOURCE SOFTWARE, 3., 2009, Kobe. **IEEE INTERNATIONAL...** Kobe: IEEE, 2009. vol. 3, p. 5. Disponível em: <https://goo.gl/mERfzH>. Acesso em: 19 jan. 2017.

RANDELL, D.; CUI, Zhan; COHN, Anthony G. A spatial logic based on regions and connection. Em: PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 3., 1992, Cambridge. **PROCEEDINGS...** Cambridge: [s.n.], 1992. vol. 92, pp. 165–176.

RANDELL, D.; WITKOWSKI, Mark. Building large composition tables via axiomatic theories. Em: INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 8., 2002, Toulouse. **PROCEEDINGS...** Toulouse: Morgan Kaufmann, 2002. pp. 26–36. Disponível em: <https://goo.gl/nXSZb7>. Acesso em: 19 jan. 2017.

RANDELL, D.; WITKOWSKI, Mark; SHANAHAN, Murray. From images to bodies: modelling and exploiting spatial occlusion and motion parallax. Em: INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE, 17., 2001, Seattle. **PROCEEDINGS...** Seattle: [s.n.], 2001. pp. 57–66. Disponível em: <https://goo.gl/zfY8MJ>. Acesso em: 19 jan. 2017.

REITER, Raymond. **Knowledge in action: logical foundations for specifying and implementing dynamical systems**. 1. ed. [S.l.]: MIT press, 2001.

SABHARWAL, Chaman L.; LEOPOLD, Jennifer L.; ELOE, Nathan. A more expressive 3D region connection calculus. Em: INTERNATIONAL CONFERENCE ON DISTRIBUTED MULTIMEDIA SYSTEMS, 2001, [S.l.]. **INTERNATIONAL...** [S.l.]: Citeseer, 2011. pp. 307–311.

SANTOS, Paulo. Reasoning about depth and motion from an observer's viewpoint. **Spatial Cognition and Computation**, [S.l.], vol. 7, pp. 133–178, 2007.

SANTOS, Paulo; SHANAHAN, Murray. Hypothesising object relations from image transitions. Em: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 15., 2002, Lyon. **PROCEEDINGS...** Lyon: [s.n.], 2002. pp. 292–296. Disponível em: <http://fei.edu.br/~psantos/santos_ecai02.pdf>. Acesso em: 19 jan. 2017.

SHANAHAN, Murray. Robotics and the common sense informatic situation. Em: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 12., 1996, Budapest. **PROCEEDINGS...** Budapest: Wiley, 1996. pp. 684–688. Disponível em: <https://goo.gl/nMrw6j>. Acesso em: 19 jan. 2017.

SOUTCHANSKI, Mikhail; SANTOS, Paulo. Reasoning about dynamic depth profiles. Em: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 18., 2008, Patras. **PROCEEDINGS...** Patras: IOS, 2008. pp. 30–34.

TAKAHASHI, Kazuko. Reasoning about relative relationships in 3D space for objects extracted from dynamic image data. Em: ECAI WORKSHOP ON SPATIO–TEMPORAL DYNAMICS, 20., 2012, Montpellier. **PROCEEDINGS...** Montpellier: IOS, 2012. pp. 45–51. Disponível em: <https://goo.gl/JSfqko>. Acesso em: 19 jan. 2017.

TRAVERSO, Paolo; GHALLAB, Malik; NAU, Dana. **Automated planning: theory and practice**. 1. ed. [S.l.]: Morgan Kauffman, 2004.

WAGNER, Thomas; HÜBNER, Kai. An egocentric qualitative spatial knowledge representation based on ordering information for physical robot navigation. In: Nardi, Daniele et. al. Em: **ROBOCUP 2004: ROBOT SOCCER WORLD CUP VIII**. [S.l.]: Springer, 2004. pp. 134–149.

ZEE, Emile Van der; SLACK, Jon Michael. **Representing direction in language and space**. 1. ed. [S.l.]: Oxford University Press on Demand, 2003. vol. 1.

APÊNDICE A – AXIOMAS DDPC

Neste apêndice estão impressos todas as fórmulas que foram omitidas no capítulo três por serem extensas demais para acompanhar o fluxo normal do texto.

Axioma de transição *extending/3*

$$
\begin{aligned}
&extending(pk(b,u,z,d),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists p,t,u',z',d')a = sense(p,loc(x_r,y_r),t) \wedge \\
&\quad pk(b,u',z',d') \in p \wedge location(R,loc(x_r,y_r),s) \wedge limitBoundary(L) \wedge \\
&\quad size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge \\
&\quad z < z' \wedge d \leq L \wedge d' \leq L \\
&\vee (\exists x,y,x_1,y_1,t,r,e,L,u',z')a = startMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&\quad location(R,loc(x_1,y_1),s) \wedge location(b,loc(x,y),s) \wedge limitBoundary(L) \wedge \\
&\quad radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_1,y_1),s) \wedge \\
&\quad depth(pk(b,u,z,d),u,loc(x_1,y_1),s) \wedge euD(loc(x,y),loc(x_r,y_r),e) \wedge \\
&\quad z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z < z' \wedge d \leq L \wedge d' \leq L \\
&\vee (\exists x_1,y_1,x_2,y_2,t,r,e,L,u',z')a = startMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \qquad (106) \\
&\quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge limitBoundary(L) \wedge \\
&\quad radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge \\
&\quad depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge euD(loc(x_2,y_2),loc(x_r,y_r),e) \wedge \\
&\quad z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z < z' \wedge d \leq L \wedge d' \leq L \\
&\vee (\neg\exists t,l_{r2},l_{b2},p',u',z',d')(a = endMove(R,loc(x_r,y_r),l_{r2},t) \\
&\qquad \vee a = endMove(b,loc(x,y),l_{b2},t) \\
&\qquad \vee a = sense(p',loc(x_r,y_r),t) \wedge pk(b,u',z',d') \in p' \wedge (z \geq z' \vee d' > L)) \wedge \\
&\qquad\quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x,y),s) \wedge \\
&\qquad\quad extending(pk(b,u,z,d),loc(x_r,y_r),s).
\end{aligned}
$$

Axioma de transição *appearing/3*

$$
\begin{aligned}
&appearing(pk(b,u,z,d),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists p,t,L,TB)a = sense(p,loc(x_r,y_r),t) \wedge pk(b,u,z,d) \in p \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge limitBoundary(L) \wedge thresholdBoundary(TB) \wedge \\
&\quad size(pk(b,u',z',d'),z',loc(x_r,y_r),s) \wedge \\
&\quad depth(pk(b,u',z',d'),d',loc(x_r,y_r),s) \wedge \\
&\quad z' < z \wedge d' > L \wedge d \leq Tb \\
&\vee (\exists x,y,x_1,y_1,t,r,e,L,TB)a = startMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge
\end{aligned}
$$

$$
\begin{aligned}
& location(R,loc(x_1,y_1),s) \wedge location(b,loc(x,y),s) \wedge limitBoundary(L) \wedge \\
& thresholdBoundary(TB) \wedge radius(b,r) \wedge \\
& size(pk(b,u'z',d'),z',loc(x_1,y_1),s) \wedge \\
& depth(pk(b,u',z',d'),u',loc(x_1,y_1),s)) \wedge \\
& euD(loc(x,y),loc(x_r,y_r),e) \wedge z = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge \\
& u = e - r \wedge z' < z \wedge d' > L \wedge d \leq TB \\
\vee\ & (\exists x_1,y_1,x_2,y_2,t,r,e,L,TB)a = startMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
& location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge limitBoundary(L) \wedge \\
& thresholdBoundary(TB) \wedge radius(b,r) \wedge \\
& size(pk(b,u'z',d'),z',loc(x_r,y_r),s) \wedge \\
& depth(pk(b,u',z',d'),u',loc(x_r,y_r),s)) \wedge \\
& euD(loc(x_2,y_2),loc(x_r,y_r),e) \wedge z = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge \\
& u = e - r \wedge z' < z \wedge d' > L \wedge d \leq TB \\
\vee\ & (\neg\exists t,l_{r2},l_{b2},p',u',z',d')(a = endMove(R,loc(x_r,y_r),l_{r2},t) \\
& \vee \quad a = endMove(b,loc(x,y),l_{b2},t) \\
& \vee \quad a = sense(p',loc(x_r,y_r),t) \wedge pk(b,u',z',d') \in p' \wedge \\
& \quad (z \geq z' \vee d' \leq L)) \wedge \\
& location(R,loc(x_r,y_r),s) \wedge location(b,loc(x,y),s) \wedge \\
& appearing(pk(b,u,z,d),loc(x_r,y_r),s).
\end{aligned}
\tag{107}
$$

Axioma de transição *shrinking/3*

$$
\begin{aligned}
& shrinking(pk(b,u,z,d),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
& \quad (\exists p,t,u',z',d',L)a = sense(p,loc(x_r,y_r),t) \wedge \\
& \quad pk(b,u',z',d') \in p \wedge location(R,loc(x_r,y_r),s) \wedge limitBoundary(L) \wedge \\
& \quad size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge \\
& \quad z > z' \wedge d \leq L \wedge d' \leq L \\
\vee\ & (\exists x,y,x_1,y_1,t,r,e,L,u',z')a = startMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
& \quad location(R,loc(x_1,y_1),s) \wedge location(b,loc(x,y),s) \wedge limitBoundary(L) \wedge \\
& \quad radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_1,y_1),s) \wedge \\
& \quad depth(pk(b,u,z,d),u,loc(x_1,y_1),s) \wedge euD(loc(x,y),loc(x_r,y_r),e) \wedge \\
& \quad z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z > z' \wedge d \leq L \wedge d' \leq L \\
\vee\ & (\exists x_1,y_1,x_2,y_2,t,r,e,L,u',z')a = startMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
& \quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge limitBoundary(L) \wedge
\end{aligned}
\tag{108}
$$

$$
\begin{aligned}
& radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge \\
& depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge euD(loc(x_2,y_2),loc(x_r,y_r),e) \wedge \\
& z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z > z' \wedge d \leq L \wedge d' \leq L \\
\vee\ & (\neg\exists t,l_{r2},l_{b2},p',u',z',d')(a = endMove(R,loc(x_r,y_r),l_{r2},t) \\
& \quad \vee a = endMove(b,loc(x,y),l_{b2},t) \\
& \quad \vee a = sense(p',loc(x_r,y_r),t) \wedge pk(b,u',z',d') \in p' \wedge (z \leq z' \vee d' > L)) \wedge \\
& \quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x,y),s) \wedge \\
& \quad shrinking(pk(b,u,z,d),loc(x_r,y_r),s).
\end{aligned}
$$

Axioma de transição *vanishing/3*

$$
\begin{aligned}
& vanishing(pk(b,u,z,d),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
& \quad (\exists p,t,L,TB,u',z',d')a = sense(p,loc(x_r,y_r),t) \wedge \\
& \quad pk(b,u',z',d') \in p \wedge location(R,loc(x_r,y_r),s) \wedge limitBoundary(L) \wedge \\
& \quad thresholdBoundary(TB) \wedge size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge \\
& \quad depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \ \wedge z > z' \wedge d > L \wedge d \leq Tb \\
\vee\ & (\exists x,y,x_1,y_1,t,r,e,L,TB,z',u')a = startMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
& \quad location(R,loc(x_1,y_1),s) \wedge location(b,loc(x,y),s) \wedge limitBoundary(L) \wedge \\
& \quad thresholdBoundary(TB) \wedge radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge \\
& \quad depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge euD(loc(x,y),loc(x_r,y_r),e) \wedge \\
& \quad z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z > z' \wedge d > L \wedge d \leq TB \\
\vee\ & (\exists x_1,y_1,x_2,y_2,t,r,e,L,TB,z',u')a = startMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \qquad (109) \\
& \quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge limitBoundary(L) \wedge \\
& \quad thresholdBoundary(TB) \wedge radius(b,r) \wedge size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge \\
& \quad depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge euD(loc(x_2,y_2),loc(x_r,y_r),e) \wedge \\
& \quad z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z > z' \wedge d > L \wedge d \leq TB \\
\vee\ & (\neg\exists t,l_{r2},l_{b2},p',u',z',d')(a = endMove(R,loc(x_r,y_r),l_{r2},t) \\
& \quad \vee a = endMove(b,loc(x,y),l_{b2},t) \\
& \quad \vee a = sense(p',loc(x_r,y_r),t) \wedge pk(b,u',z',d') \in p' \wedge (z \leq z' \vee d' \leq L)) \wedge \\
& \quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x,y),s) \wedge \\
& \quad vanishing(pk(b,u,z,d),loc(x_r,y_r),s).
\end{aligned}
$$

Axioma de transição *peak_static/3*

$$
peak_static(pk(b,u,z,d),loc(x_r,y_r),do(a,s)) \leftrightarrow
$$

$$
\begin{aligned}
&(\exists p,t,u',z',d')a = sense(p,loc(x_r,y_r),t) \wedge pk(b,u',z',d') \in p \wedge \\
&location(R,loc(x_r,y_r),s) \wedge size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge \\
&depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge dist(pk(b,u,z,d),d,loc(x_r,y_r),s) \wedge \\
&z = z' \wedge u = u' \wedge d = d' \\
\vee\ &(\exists x,y,x_1,y_1,t,r,e,z',u',d')a = startMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge \\
&location(R,loc(x_1,y_1),s) \wedge location(b,loc(x,y),s) \wedge radius(b,r) \wedge \\
&size(pk(b,u,z,d),z,loc(x_1,y_1),s) \wedge depth(pk(b,u,z,d),u,loc(x_1,y_1),s) \wedge \\
&dist(pk(b,u,z,d),d,loc(x_1,y_1),s) \wedge euD(loc(x,y),loc(x_r,y_r),e) \wedge \\
&z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z = z' \wedge u = u' \wedge d = d' \\
\vee\ &(\exists x_1,y_1,x_2,y_2,t,r,e,z',u',d')a = startMove(b,loc(x_1,y_1),loc(x_2,y_2),t) \wedge \\
&location(R,loc(x_r,y_r),s) \wedge location(b,loc(x_1,y_1),s) \wedge radius(b,r) \wedge \\
&size(pk(b,u,z,d),z,loc(x_r,y_r),s) \wedge depth(pk(b,u,z,d),u,loc(x_r,y_r),s) \wedge \\
&dist(pk(b,u,z,d),d,loc(x_r,y_r),s) \wedge euD(loc(x_2,y_2),loc(x_r,y_r),e) \wedge \\
&z' = 2 \cdot \arcsin(r/e) \cdot 180/\pi \wedge u' = e - r \wedge z = z' \wedge u = u' \wedge d = d' \\
\vee\ &(\neg\exists t,l_{r2},l_{b2},p',u',z',d')(a = endMove(R,loc(x_r,y_r),l_{r2},t) \\
&\quad \vee a = endMove(b,loc(x,y),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
&\quad pk(b,u',z',d') \in p' \wedge (z \neq z' \vee u \neq u' \vee d \neq d')) \wedge \\
&\quad location(R,loc(x_r,y_r),s) \wedge location(b,loc(x,y),s) \wedge \\
&\quad peak_static(pk(b,u,z,d),loc(x_r,y_r),s).
\end{aligned}
\tag{110}
$$

Axioma de transição *approaching/4*

$$
\begin{aligned}
&approaching(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists p,t,\Delta,d')a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge \\
&\quad pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\quad ((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge \\
&\quad thresholdUnion(\Delta) \wedge d' < d \wedge d' \geq \Delta \\
&\vee\ (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,e_b,z'_a,z'_b,\gamma) \\
&\quad a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
&\quad radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge \\
&\quad euD(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge euD(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge \\
&\quad z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge
\end{aligned}
$$

$$
\begin{aligned}
& angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,z'_a,\gamma) \\
& a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& euD(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \qquad (111) \\
& angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_b,z'_b,\gamma) \\
& a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& euD(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a),loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
\vee\ & (\neg\exists p',t,\Delta,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b) \\
& (a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(b_a,loc(x_a,y_a),l_{a2},t) \\
& \quad \vee a = endMove(b_b,loc(x_b,y_b),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
& \quad pk(b_a,u'_a,z'_a,d'_a) \in p' \wedge pk(b_b,u'_b,z'_b,d'_b) \in p' \wedge \\
& \quad ((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge \\
& \quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& \quad thresholdUnion(\Delta) \wedge (d' \geq d \vee d' < \Delta)) \wedge \\
& location(R,loc(x_r,y_r),s) \wedge location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
& approaching(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s).
\end{aligned}
$$

Axioma de transição *coalescing/4*

$$
\begin{aligned}
& coalescing(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
& \quad (\exists p,t,\Delta,d',x_a,y_a,x_b,y_b,e_a,e_b) a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge \\
& \quad pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge \\
& \quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& \quad ((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge \\
& \quad thresholdUnion(\Delta) \wedge d' < d \wedge d' < \Delta \wedge
\end{aligned}
$$

$$
\begin{aligned}
& thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge \\
& location(b_a, loc(x_a,y_a), s) \wedge location(b_b, loc(x_b,y_b), s) \wedge \\
& euD(loc(x_a, y_a), loc(x_r, y_r), e_a) \wedge euD(loc(x_b, y_b), loc(x_r, y_r), e_b) \wedge \\
& (e_a >= e_b \wedge d > z_a \cdot (-1) \vee e_b > e_a \wedge d > z_b \cdot (-1)) \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,e_b,z'_a,z'_b,\gamma) \\
& a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge \\
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge \\
& eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge \\
& z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' < \Delta \wedge \\
& thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1)) \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,z'_a,\gamma) \\
& a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' < \Delta \wedge \\
& thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z_b \cdot (-1)) \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_b,z'_b,\gamma) \\
& a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdUnion(\Delta) \wedge d' < \Delta \wedge \\
& thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1)) \\
\vee\ & (\neg\exists p',t,\Delta,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b) \\
& (a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(b_a,loc(x_a,y_a),l_{a2},t)
\end{aligned}
\tag{112}
$$

$\vee\ a = endMove(b_b,loc(x_b,y_b),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge$

$pk(b_a,u'_a,z'_a,d'_a) \in p' \wedge pk(b_b,u'_b,z'_b,d'_b) \in p' \wedge$

$((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge$

$dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge$

$thresholdUnion(\Delta) \wedge (d' \geq d \vee d' \geq \Delta)) \wedge$

$location(R,loc(x_r,y_r),s) \wedge location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge$

$coalescing(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s).$

Axioma de transição *splitting/4*

$splitting(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow$

$(\exists p,t,\Delta,d')a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge$

$pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge$

$dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge$

$((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge$

$thresholdUnion(\Delta) \wedge d' > d \wedge d' < \Delta \wedge$

$thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge$

$location(b_a, loc(x_a,y_a), s) \wedge location(b_b, loc(x_b,y_b), s) \wedge$

$euD(loc(x_a, y_a), loc(x_r, y_r), e_a) \wedge euD(loc(x_b, y_b), loc(x_r, y_r), e_b) \wedge$

$(e_a >= e_b \wedge d > z_a \cdot (-1) \vee e_b > e_a \wedge d > z_b \cdot (-1))$

$\vee\ (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,e_b,z'_a,z'_b,\gamma)$

$a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge$

$location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge$

$radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge$

$eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge$

$z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge$

$angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge$

$d' = \gamma - (z'_a/2) - (z'_b/2) \wedge d' > d \wedge thresholdUnion(\Delta) \wedge d' < \Delta \wedge$

$thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge$

$(e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1))$

$\vee\ (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,z'_a,\gamma)$

$a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge$

$location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge$

$radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge$

$$
\begin{aligned}
& eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z_b/2) \wedge d' > d \wedge thresholdUnion(\Delta) \wedge d' < \Delta \wedge \\
& thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z_b \cdot (-1)) \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_b,z'_b,\gamma) \\
& a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z_a/2) - (z'_b/2) \wedge d' > d \wedge thresholdUnion(\Delta) \wedge d' < \Delta \wedge \\
& thresholdHiding(\lambda) \wedge d' \geq \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1)) \\
\vee\ & (\neg\exists p',t,\Delta,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b) \\
& (a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(b_a,loc(x_a,y_a),l_{a2},t) \\
& \quad \vee a = endMove(b_b,loc(x_b,y_b),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
& \qquad pk(b_a,u'_a,z'_a,d'_a) \in p' \wedge pk(b_b,u'_b,z'_b,d'_b) \in p' \wedge \\
& \qquad ((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge \\
& \qquad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& \qquad thresholdUnion(\Delta) \wedge (d' \leq d \vee d' \geq \Delta)) \wedge \\
& \quad location(R,loc(x_r,y_r),s) \wedge location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
& \quad splitting(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s).
\end{aligned} \tag{113}
$$

Axioma de transição *receding/4*

$$
\begin{aligned}
& receding(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
& \quad (\exists p,t,\Delta,d')a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge \\
& \quad pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge \\
& \quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& \quad ((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge \\
& \quad thresholdUnion(\Delta) \wedge d' > d \wedge d' \geq \Delta \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,e_b,z'_a,z'_b,\gamma) \\
& \quad a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge
\end{aligned}
$$

$$
\begin{aligned}
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge \\
& eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge \\
& z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z'_b/2) \wedge d' > d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,z'_a,\gamma) \\
& a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z_b/2) \wedge d' > d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_b,z'_b,\gamma) \\
& a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z_a/2) - (z'_b/2) \wedge d' > d \wedge thresholdUnion(\Delta) \wedge d' \geq \Delta \\
\vee\ & (\neg\exists p',t,\Delta,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b) \\
& (a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(b_a,loc(x_a,y_a),l_{a2},t) \\
& \quad \vee a = endMove(b_b,loc(x_b,y_b),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
& \qquad pk(b_a,u'_a,z'_a,d'_a) \in p' \wedge pk(b_b,u'_b,z'_b,d'_b) \in p' \wedge \\
& \qquad ((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge \\
& \qquad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& \qquad thresholdUnion(\Delta) \wedge (d' \leq d \vee d' < \Delta)) \wedge \\
& \quad location(R,loc(x_r,y_r),s) \wedge location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
& \quad receding(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s).
\end{aligned}
\tag{114}
$$

Axioma de transição *hiding/4*

$$
\begin{aligned}
& hiding(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
& \quad (\exists p,t,\Delta,d',x_a,y_a,x_b,y_b,e_a,e_b) a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge
\end{aligned}
$$

$$
\begin{aligned}
& pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge \\
& dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& ((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge \\
& d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
& location(b_a, loc(x_a,y_a), s) \wedge location(b_b, loc(x_b,y_b), s) \wedge \\
& euD(loc(x_a, y_a), loc(x_r, y_r), e_a) \wedge euD(loc(x_b, y_b), loc(x_r, y_r), e_b) \wedge \\
& (e_a >= e_b \wedge d > z_a \cdot (-1) \vee e_b > e_a \wedge d > z_b \cdot (-1)) \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,e_b,z'_a,z'_b,\gamma) \\
& a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge \\
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge \\
& eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge \\
& z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1)) \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_a,z'_a,\gamma) \\
& a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z'_a/2) - (z_b/2) \wedge d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z'_a \cdot (-1) \vee e_b > e_a \wedge d_a > z_b \cdot (-1)) \\
\vee\ & (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,\Delta,d,d',e_b,z'_b,\gamma) \\
& a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
& location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge \\
& radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
& eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
& angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge \\
& d' = \gamma - (z_a/2) - (z'_b/2) \wedge d' < d \wedge thresholdHiding(\lambda) \wedge d' < \lambda \wedge \\
& (e_a >= e_b \wedge d_a > z_a \cdot (-1) \vee e_b > e_a \wedge d_a > z'_b \cdot (-1)) \\
\vee\ & (\neg\exists p',t,\Delta,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b) \\
& (a = endMove(R,loc(x_r,y_r),l_{r2},t) \vee a = endMove(b_a,loc(x_a,y_a),l_{a2},t)
\end{aligned}
\tag{115}
$$

$$
\begin{aligned}
&\vee\ a = endMove(b_b,loc(x_b,y_b),l_{b2},t) \vee a = sense(p',loc(x_r,y_r),t) \wedge \\
&\quad pk(b_a,u'_a,z'_a,d'_a) \in p' \wedge pk(b_b,u'_b,z'_b,d'_b) \in p' \wedge \\
&\quad ((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge \\
&\quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\quad thresholdUnion(\Delta) \wedge (d' \geq d \vee d' \geq \Delta)) \wedge \\
&location(R,loc(x_r,y_r),s) \wedge location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge \\
&hiding(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),s).
\end{aligned}
$$

Axioma de transição *two_peak_static/4*

$$
\begin{aligned}
&two_peak_static(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),loc(x_r,y_r),do(a,s)) \leftrightarrow \\
&\quad (\exists p,t,d')a = sense(p,loc(x_r,y_r),t) \wedge pk(b_a,u_a,z_a,d_a) \in p \wedge \\
&\quad pk(b_b,u_b,z_b,d_b) \in p \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\quad ((d_b > d_a + z_a \wedge d' = d_b - d_a - z_a) \vee (d_a > d_b + z_b \wedge d' = d_a - d_b - z_b)) \wedge \\
&\quad d' = d \\
&\vee\ (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,d,d',e_a,e_b,z'_a,z'_b,\gamma) \\
&\quad a = endMove(R,loc(x_1,y_1),loc(x_r,y_r),t) \wedge location(R,loc(x_1,y_1),s) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
&\quad radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_1,y_1),s) \wedge \\
&\quad eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge eud(loc(x_b,y_b),loc(x_r,y_r),e_b) \wedge \\
&\quad z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
&\quad angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge d' = \gamma - (z'_a/2) - (z'_b/2) \wedge \\
&\quad d' = d \\
&\vee\ (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,d,d',e_a,z'_a,\gamma) \\
&\quad a = endMove(b_a,loc(x_1,y_1),loc(x_a,y_a),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad location(b_a,loc(x_1,y_1),s) \wedge location(b_b,loc(x_b,y_b),s) \wedge radius(b_a,r_a) \wedge \\
&\quad radius(b_b,r_b) \wedge dist(pk(b_a,u_a,z_a,d_a),pk(b_b,u_b,z_b,d_b),d,loc(x_r,y_r),s) \wedge \\
&\quad eud(loc(x_a,y_a),loc(x_r,y_r),e_a) \wedge z'_a = 2 \cdot \arcsin(r_a/e_a) \cdot 180/\pi \wedge \\
&\quad angleV(loc(x_a,y_a,loc(x_b,y_b),loc(x_r,y_r),\gamma) \wedge d' = \gamma - (z'_a/2) - (z_b/2) \wedge \qquad (116) \\
&\quad d' = d \\
&\vee\ (\exists x_1,y_1,x_a,y_a,x_b,y_b,r_a,r_b,t,d,d',e_b,z'_b,\gamma) \\
&\quad a = endMove(b_b,loc(x_1,y_1),loc(x_b,y_b),t) \wedge location(R,loc(x_r,y_r),s) \wedge \\
&\quad location(b_a,loc(x_a,y_a),s) \wedge location(b_b,loc(x_1,y_1),s) \wedge radius(b_a,r_a) \wedge
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{radius}(b_b,r_b) \wedge \mathit{dist}(\mathit{pk}(b_a,u_a,z_a,d_a),\mathit{pk}(b_b,u_b,z_b,d_b),d,\mathit{loc}(x_r,y_r),s) \wedge \\
&\mathit{eud}(\mathit{loc}(x_b,y_b),\mathit{loc}(x_r,y_r),e_b) \wedge z'_b = 2 \cdot \arcsin(r_b/e_b) \cdot 180/\pi \wedge \\
&\mathit{angleV}(\mathit{loc}(x_a,y_a,\mathit{loc}(x_b,y_b),\mathit{loc}(x_r,y_r),\gamma) \wedge d' = \gamma - (z_a/2) - (z'_b/2) \wedge \\
&d' = d \\
\vee\ &(\neg\exists p',t,x_a,y_a,x_b,y_b,d,d',l_{r2},l_{a2},l_{b2},u'_a,z'_a,d'_a,u'_b,z'_b,d'_b) \\
&\quad \Big(a = \mathit{endMove}(R,\mathit{loc}(x_r,y_r),l_{r2},t) \vee a = \mathit{endMove}(b_a,\mathit{loc}(x_a,y_a),l_{a2},t) \\
&\quad \vee a = \mathit{endMove}(b_b,\mathit{loc}(x_b,y_b),l_{b2},t) \vee a = \mathit{sense}(p',\mathit{loc}(x_r,y_r),t) \wedge \\
&\qquad \mathit{pk}(b_a,u'_a,z'_a,d'_a) \in p' \wedge \mathit{pk}(b_b,u'_b,z'_b,d'_b) \in p' \wedge \\
&\qquad ((d'_a > d'_b + z'_b \wedge d' = d'_a - d'_b - z'_b) \vee (d'_b > d'_a + z'_a \wedge d' = d'_b - d'_a - z'_a)) \wedge \\
&\qquad \mathit{dist}(\mathit{pk}(b_a,u_a,z_a,d_a),\mathit{pk}(b_b,u_b,z_b,d_b),d,\mathit{loc}(x_r,y_r),s) \wedge (d' > d \vee d' < \Delta)\Big) \wedge \\
&\mathit{location}(R,\mathit{loc}(x_r,y_r),s) \wedge \mathit{location}(b_a,\mathit{loc}(x_a,y_a),s) \wedge \\
&\mathit{location}(b_b,\mathit{loc}(x_b,y_b),s) \wedge \\
&\mathit{two_peak_static}(\mathit{pk}(b_a,u_a,z_a,d_a),\mathit{pk}(b_b,u_b,z_b,d_b),\mathit{loc}(x_r,y_r),s).
\end{aligned}
$$

APÊNDICE B – TABELA DE TESTES SENTIDO ANTI-HORÁRIO

Tabela 3 – Resultados dos testes de transição para pares de picos

(continua)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(0.0,8.0),loc(-1.83,7.93),20)	1	12.454	13.351	72.034	false	false	false	false	**true**	false
do(endMove(cpto,loc(0.0,8.0),loc(-1.83,7.93),21)	2	14.605	11.798	72.465	false	false	false	false	false	**true**
do(startPan(7.5,22)	3	14.605	11.798	72.465	false	false	false	false	false	**true**
do(endPan(7.5,23)	4	14.605	11.798	72.465	false	false	false	false	false	**true**
do(startMove(cpto,loc(-1.83,7.93),loc(-3.62,7.73),24)	5	14.605	11.798	72.465	**true**	false	false	false	false	false
do(endMove(cpto,loc(-1.83,7.93),loc(-3.62,7.73),25)	6	17.291	10.493	68.663	false	false	false	false	false	**true**
do(startPan(7.5,26)	7	17.291	10.493	68.663	false	false	false	false	false	**true**
do(endPan(7.5,27)	8	17.291	10.493	68.663	false	false	false	false	false	**true**
do(startMove(cpto,loc(-3.62,7.73),loc(-5.36,7.39),28)	9	17.291	10.493	68.663	**true**	false	false	false	false	false
do(endMove(cpto,loc(-3.62,7.73),loc(-5.36,7.39),29)	10	20.448	9.425	60.321	false	false	false	false	false	**true**
do(startPan(7.5,30)	11	20.448	9.425	60.321	false	false	false	false	false	**true**
do(endPan(7.5,31)	12	20.448	9.425	60.321	false	false	false	false	false	**true**
do(startMove(cpto,loc(-5.36,7.39),loc(-7.0,6.93),32)	13	20.448	9.425	60.321	**true**	false	false	false	false	false
do(endMove(cpto,loc(-5.36,7.39),loc(-7.0,6.93),33)	14	23.406	8.569	47.715	false	false	false	false	false	**true**
do(startPan(7.5,34)	15	23.406	8.569	47.715	false	false	false	false	false	**true**
do(endPan(7.5,35)	16	23.406	8.569	47.715	false	false	false	false	false	**true**
do(startMove(cpto,loc(-7.0,6.93),loc(-8.52,6.35),36)	17	23.406	8.569	47.715	**true**	false	false	false	false	false
do(endMove(cpto,loc(-7.0,6.93),loc(-8.52,6.35),37)	18	25.068	7.887	32.672	false	false	false	false	false	**true**
do(startPan(7.5,38)	19	25.068	7.887	32.672	false	false	false	false	false	**true**
do(endPan(7.5,39)	20	25.068	7.887	32.672	false	false	false	false	false	**true**
do(startMove(cpto,loc(-8.52,6.35),loc(-9.9,5.66),40)	21	25.068	7.887	32.672	**true**	false	false	false	false	false
do(endMove(cpto,loc(-8.52,6.35),loc(-9.9,5.66),41)	22	24.731	7.345	18.203	false	false	false	false	false	**true**
do(startPan(7.5,42)	23	24.731	7.345	18.203	false	false	false	false	false	**true**
do(endPan(7.5,43)	24	24.731	7.345	18.203	false	false	false	false	false	**true**
do(startMove(cpto,loc(-9.9,5.66),loc(-11.11,4.87),44)	25	24.731	7.345	18.203	false	**true**	false	false	false	false
do(endMove(cpto,loc(-9.9,5.66),loc(-11.11,4.87),45)	26	23.014	6.921	6.451	false	false	false	false	false	**true**
do(startPan(7.5,46)	27	23.014	6.921	6.451	false	false	false	false	false	**true**
do(endPan(7.5,47)	28	23.014	6.921	6.451	false	false	false	false	false	**true**

Tabela 3 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(-11.11,4.87),loc(-12.12,4.0),48)	29	23.014	6.921	6.451	false	false	**true**	false	false	false
do(endMove(cpto,loc(-11.11,4.87),loc(-12.12,4.0),49)	30	20.964	6.597	-2.383	false	false	false	false	false	**true**
do(startPan(7.5,50)	31	20.964	6.597	-2.383	false	false	false	false	false	**true**
do(endPan(7.5,51)	32	20.964	6.597	-2.383	false	false	false	false	false	**true**
do(startMove(cpto,loc(-12.12,4.0),loc(-12.93,3.06),52)	33	20.964	6.597	-2.383	false	false	**true**	false	false	false
do(endMove(cpto,loc(-12.12,4.0),loc(-12.93,3.06),53)	34	19.111	6.353	-9.151	false	false	false	false	false	**true**
do(startPan(7.5,54)	35	19.111	6.353	-9.151	false	false	false	false	false	**true**
do(endPan(7.5,55)	36	19.111	6.353	-9.151	false	false	false	false	false	**true**
do(startMove(cpto,loc(-12.93,3.06),loc(-13.52,2.07),56)	37	19.111	6.353	-9.151	false	false	false	false	false	false
do(endMove(cpto,loc(-12.93,3.06),loc(-13.52,2.07),57)	38	17.644	6.180	-9.221	false	false	false	false	false	**true**
do(startPan(7.5,58)	39	17.644	6.180	-9.221	false	false	false	false	false	**true**
do(endPan(7.5,59)	40	17.644	6.180	-9.221	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.52,2.07),loc(-13.88,1.04),60)	41	17.644	6.180	-9.221	false	false	false	false	false	false
do(endMove(cpto,loc(-13.52,2.07),loc(-13.88,1.04),61)	42	16.553	6.072	-3.248	false	false	false	false	false	**true**
do(startPan(7.5,62)	43	16.553	6.072	-3.248	false	false	false	false	false	**true**
do(endPan(7.5,63)	44	16.553	6.072	-3.248	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.88,1.04),loc(-14.0,-0.0),64)	45	16.553	6.072	-3.248	false	false	false	false	false	false
do(endMove(cpto,loc(-13.88,1.04),loc(-14.0,-0.0),65)	46	15.790	6.026	2.025	false	false	false	false	false	**true**
do(startPan(7.5,66)	47	15.790	6.026	2.025	false	false	false	false	false	**true**
do(endPan(7.5,67)	48	15.790	6.026	2.025	false	false	false	false	false	**true**
do(startMove(cpto,loc(-14.0,-0.0),loc(-13.88,-1.04),68)	49	15.790	6.026	2.025	false	false	false	**true**	false	false
do(endMove(cpto,loc(-14.0,-0.0),loc(-13.88,-1.04),69)	50	15.280	6.037	7.013	false	false	false	false	false	**true**
do(startPan(7.5,70)	51	15.280	6.037	7.013	false	false	false	false	false	**true**
do(endPan(7.5,71)	52	15.280	6.037	7.013	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.88,-1.04),loc(-13.52,-2.07),72)	53	15.280	6.037	7.013	false	false	false	**true**	false	false
do(endMove(cpto,loc(-13.88,-1.04),loc(-13.52,-2.07),73)	54	14.951	6.107	12.032	false	false	false	false	false	**true**
do(startPan(7.5,74)	55	14.951	6.107	12.032	false	false	false	false	false	**true**
do(endPan(7.5,75)	56	14.951	6.107	12.032	false	false	false	false	false	**true**
do(startMove(cpto,loc(-13.52,-2.07),loc(-12.93,-3.06),76)	57	14.951	6.107	12.032	false	false	false	false	**true**	false
do(endMove(cpto,loc(-13.52,-2.07),loc(-12.93,-3.06),77)	58	14.740	6.236	17.227	false	false	false	false	false	**true**
do(startPan(7.5,78)	59	14.740	6.236	17.227	false	false	false	false	false	**true**
do(endPan(7.5,79)	60	14.740	6.236	17.227	false	false	false	false	false	**true**

Tabela 3 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(-12.93,-3.06),loc(-12.12,-4.0),80)	61	14.740	6.236	17.227	false	false	false	false	**true**	false
do(endMove(cpto,loc(-12.93,-3.06),loc(-12.12,-4.0),81)	62	14.567	6.428	22.746	false	false	false	false	false	**true**
do(startPan(7.5,82)	63	14.567	6.428	22.746	false	false	false	false	false	**true**
do(endPan(7.5,83)	64	14.567	6.428	22.746	false	false	false	false	false	**true**
do(startMove(cpto,loc(-12.12,-4.0),loc(-11.11,-4.87),84)	65	14.567	6.428	22.746	false	false	false	false	**true**	false
do(endMove(cpto,loc(-12.12,-4.0),loc(-11.11,-4.87),85)	66	14.351	6.687	28.570	false	false	false	false	false	**true**
do(startPan(7.5,86)	67	14.351	6.687	28.570	false	false	false	false	false	**true**
do(endPan(7.5,87)	68	14.351	6.687	28.570	false	false	false	false	false	**true**
do(startMove(cpto,loc(-11.11,-4.87),loc(-9.9,-5.66),88)	69	14.351	6.687	28.570	false	false	false	false	**true**	false
do(endMove(cpto,loc(-11.11,-4.87),loc(-9.9,-5.66),89)	70	14.026	7.026	34.655	false	false	false	false	false	**true**
do(startPan(7.5,90)	71	14.026	7.026	34.655	false	false	false	false	false	**true**
do(endPan(7.5,91)	72	14.026	7.026	34.655	false	false	false	false	false	**true**
do(startMove(cpto,loc(-9.9,-5.66),loc(-8.52,-6.35),92)	73	14.026	7.026	34.655	false	false	false	false	**true**	false
do(endMove(cpto,loc(-9.9,-5.66),loc(-8.52,-6.35),93)	74	13.533	7.452	40.660	false	false	false	false	false	**true**
do(startPan(7.5,94)	75	13.533	7.452	40.660	false	false	false	false	false	**true**
do(endPan(7.5,95)	76	13.533	7.452	40.660	false	false	false	false	false	**true**
do(startMove(cpto,loc(-8.52,-6.35),loc(-7.0,-6.93),96)	77	13.533	7.452	40.660	false	false	false	false	**true**	false
do(endMove(cpto,loc(-8.52,-6.35),loc(-7.0,-6.93),97)	78	12.859	7.973	46.126	false	false	false	false	false	**true**
do(startPan(7.5,98)	79	12.859	7.973	46.126	false	false	false	false	false	**true**
do(endPan(7.5,99)	80	12.859	7.973	46.126	false	false	false	false	false	**true**
do(startMove(cpto,loc(-7.0,-6.93),loc(-5.36,-7.39),100)	81	12.859	7.973	46.126	false	false	false	false	**true**	false
do(endMove(cpto,loc(-7.0,-6.93),loc(-5.36,-7.39),101)	82	12.044	8.604	50.581	false	false	false	false	false	**true**
do(startPan(7.5,102)	83	12.044	8.604	50.581	false	false	false	false	false	**true**
do(endPan(7.5,103)	84	12.044	8.604	50.581	false	false	false	false	false	**true**
do(startMove(cpto,loc(-5.36,-7.39),loc(-3.62,-7.73),104)	85	12.044	8.604	50.581	false	false	false	false	**true**	false
do(endMove(cpto,loc(-5.36,-7.39),loc(-3.62,-7.73),105)	86	11.143	9.351	53.546	false	false	false	false	false	**true**
do(startPan(7.5,106)	87	11.143	9.351	53.546	false	false	false	false	false	**true**
do(endPan(7.5,107)	88	11.143	9.351	53.546	false	false	false	false	false	**true**
do(startMove(cpto,loc(-3.62,-7.73),loc(-1.83,-7.93),108)	89	11.143	9.351	53.546	false	false	false	false	**true**	false
do(endMove(cpto,loc(-3.62,-7.73),loc(-1.83,-7.93),109)	90	10.249	10.206	54.686	false	false	false	false	false	**true**
do(startPan(7.5,110)	91	10.249	10.206	54.686	false	false	false	false	false	**true**
do(endPan(7.5,111)	92	10.249	10.206	54.686	false	false	false	false	false	**true**

Tabela 3 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(-1.83,-7.93),loc(0.0,-8.0),112)	93	10.249	10.206	54.686	**true**	false	false	false	false	false
do(endMove(cpto,loc(-1.83,-7.93),loc(0.0,-8.0),113)	94	9.398	11.148	53.774	false	false	false	false	false	**true**
do(startPan(7.5,114)	95	9.398	11.148	53.774	false	false	false	false	false	**true**
do(endPan(7.5,115)	96	9.398	11.148	53.774	false	false	false	false	false	**true**
do(startMove(cpto,loc(0.0,-8.0),loc(1.83,-7.93),116)	97	9.398	11.148	53.774	**true**	false	false	false	false	false
do(endMove(cpto,loc(0.0,-8.0),loc(1.83,-7.93),117)	98	8.632	12.115	50.815	false	false	false	false	false	**true**
do(startPan(7.5,118)	99	8.632	12.115	50.815	false	false	false	false	false	**true**
do(endPan(7.5,119)	100	8.632	12.115	50.815	false	false	false	false	false	**true**
do(startMove(cpto,loc(1.83,-7.93),loc(3.62,-7.73),120)	101	8.632	12.115	50.815	**true**	false	false	false	false	false
do(endMove(cpto,loc(1.83,-7.93),loc(3.62,-7.73),121)	102	7.962	12.993	46.009	false	false	false	false	false	**true**
do(startPan(7.5,122)	103	7.962	12.993	46.009	false	false	false	false	false	**true**
do(endPan(7.5,123)	104	7.962	12.993	46.009	false	false	false	false	false	**true**
do(startMove(cpto,loc(3.62,-7.73),loc(5.36,-7.39),124)	105	7.962	12.993	46.009	**true**	false	false	false	false	false
do(endMove(cpto,loc(3.62,-7.73),loc(5.36,-7.39),125)	106	7.388	13.678	39.786	false	false	false	false	false	**true**
do(startPan(7.5,126)	107	7.388	13.678	39.786	false	false	false	false	false	**true**
do(endPan(7.5,127)	108	7.388	13.678	39.786	false	false	false	false	false	**true**
do(startMove(cpto,loc(5.36,-7.39),loc(7.0,-6.93),128)	109	7.388	13.678	39.786	**true**	false	false	false	false	false
do(endMove(cpto,loc(5.36,-7.39),loc(7.0,-6.93),129)	110	6.905	14.047	32.837	false	false	false	false	false	**true**
do(startPan(7.5,130)	111	6.905	14.047	32.837	false	false	false	false	false	**true**
do(endPan(7.5,131)	112	6.905	14.047	32.837	false	false	false	false	false	**true**
do(startMove(cpto,loc(7.0,-6.93),loc(8.52,-6.35),132)	113	6.905	14.047	32.837	**true**	false	false	false	false	false
do(endMove(cpto,loc(7.0,-6.93),loc(8.52,-6.35),133)	114	6.506	14.097	25.828	false	false	false	false	false	**true**
do(startPan(7.5,134)	115	6.506	14.097	25.828	false	false	false	false	false	**true**
do(endPan(7.5,135)	116	6.506	14.097	25.828	false	false	false	false	false	**true**
do(startMove(cpto,loc(8.52,-6.35),loc(9.9,-5.66),136)	117	6.506	14.097	25.828	**true**	false	false	false	false	false
do(endMove(cpto,loc(8.52,-6.35),loc(9.9,-5.66),137)	118	6.179	13.893	19.238	false	false	false	false	false	**true**
do(startPan(7.5,138)	119	6.179	13.893	19.238	false	false	false	false	false	**true**
do(endPan(7.5,139)	120	6.179	13.893	19.238	false	false	false	false	false	**true**
do(startMove(cpto,loc(9.9,-5.66),loc(11.11,-4.87),140)	121	6.179	13.893	19.238	false	**true**	false	false	false	false
do(endMove(cpto,loc(9.9,-5.66),loc(11.11,-4.87),141)	122	5.919	13.556	13.341	false	false	false	false	false	**true**
do(startPan(7.5,142)	123	5.919	13.556	13.341	false	false	false	false	false	**true**
do(endPan(7.5,143)	124	5.919	13.556	13.341	false	false	false	false	false	**true**

Tabela 3 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(11.11,-4.87),loc(12.12,-4.0),144)	125	5.919	13.556	13.341	false	**true**	false	false	false	false
do(endMove(cpto,loc(11.11,-4.87),loc(12.12,-4.0),145)	126	5.721	13.200	8.196	false	false	false	false	false	**true**
do(startPan(7.5,146)	127	5.721	13.200	8.196	false	false	false	false	false	**true**
do(endPan(7.5,147)	128	5.721	13.200	8.196	false	false	false	false	false	**true**
do(startMove(cpto,loc(12.12,-4.0),loc(12.93,-3.06),148)	129	5.721	13.200	8.196	false	false	**true**	false	false	false
do(endMove(cpto,loc(12.12,-4.0),loc(12.93,-3.06),149)	130	5.575	12.890	3.633	false	false	false	false	false	**true**
do(startPan(7.5,150)	131	5.575	12.890	3.633	false	false	false	false	false	**true**
do(endPan(7.5,151)	132	5.575	12.890	3.633	false	false	false	false	false	**true**
do(startMove(cpto,loc(12.93,-3.06),loc(13.52,-2.07),152)	133	5.575	12.890	3.633	false	false	**true**	false	false	false
do(endMove(cpto,loc(12.93,-3.06),loc(13.52,-2.07),153)	134	5.480	12.679	-.483	false	false	false	false	false	**true**
do(startPan(7.5,154)	135	5.480	12.679	-.483	false	false	false	false	false	**true**
do(endPan(7.5,155)	136	5.480	12.679	-.483	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.52,-2.07),loc(13.88,-1.04),156)	137	5.480	12.679	-.483	false	false	**true**	false	false	false
do(endMove(cpto,loc(13.52,-2.07),loc(13.88,-1.04),157)	138	5.433	12.602	-4.363	false	false	false	false	false	**true**
do(startPan(7.5,158)	139	5.433	12.602	-4.363	false	false	false	false	false	**true**
do(endPan(7.5,159)	140	5.433	12.602	-4.363	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.88,-1.04),loc(14.0,0.0),160)	141	5.433	12.602	-4.363	false	false	**true**	false	false	false
do(endMove(cpto,loc(13.88,-1.04),loc(14.0,0.0),161)	142	5.434	12.680	-8.157	false	false	false	false	false	**true**
do(startPan(7.5,162)	143	5.434	12.680	-8.157	false	false	false	false	false	**true**
do(endPan(7.5,163)	144	5.434	12.680	-8.157	false	false	false	false	false	**true**
do(startMove(cpto,loc(14.0,0.0),loc(13.88,1.04),164)	145	5.434	12.680	-8.157	false	false	false	false	false	false
do(endMove(cpto,loc(14.0,0.0),loc(13.88,1.04),165)	146	5.484	12.932	-6.318	false	false	false	false	false	**true**
do(startPan(7.5,166)	147	5.484	12.932	-6.318	false	false	false	false	false	**true**
do(endPan(7.5,167)	148	5.484	12.932	-6.318	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.88,1.04),loc(13.52,2.07),168)	149	5.484	12.932	-6.318	false	false	false	false	false	false
do(endMove(cpto,loc(13.88,1.04),loc(13.52,2.07),169)	150	5.587	13.375	-2.518	false	false	false	false	false	**true**
do(startPan(7.5,170)	151	5.587	13.375	-2.518	false	false	false	false	false	**true**
do(endPan(7.5,171)	152	5.587	13.375	-2.518	false	false	false	false	false	**true**
do(startMove(cpto,loc(13.52,2.07),loc(12.93,3.06),172)	153	5.587	13.375	-2.518	false	false	false	false	false	false
do(endMove(cpto,loc(13.52,2.07),loc(12.93,3.06),173)	154	5.744	14.021	1.635	false	false	false	false	false	**true**
do(startPan(7.5,174)	155	5.744	14.021	1.635	false	false	false	false	false	**true**
do(endPan(7.5,175)	156	5.744	14.021	1.635	false	false	false	false	false	**true**

Tabela 3 – Resultados dos testes de transição para pares de picos

(continuação)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(12.93,3.06),loc(12.12,4.0),176)	157	5.744	14.021	1.635	false	false	false	**true**	false	false
do(endMove(cpto,loc(12.93,3.06),loc(12.12,4.0),177)	158	5.963	14.873	6.458	false	false	false	false	false	**true**
do(startPan(7.5,178)	159	5.963	14.873	6.458	false	false	false	false	false	**true**
do(endPan(7.5,179)	160	5.963	14.873	6.458	false	false	false	false	false	**true**
do(startMove(cpto,loc(12.12,4.0),loc(11.11,4.87),180)	161	5.963	14.873	6.458	false	false	false	**true**	false	false
do(endMove(cpto,loc(12.12,4.0),loc(11.11,4.87),181)	162	6.253	15.895	12.271	false	false	false	false	false	**true**
do(startPan(7.5,182)	163	6.253	15.895	12.271	false	false	false	false	false	**true**
do(endPan(7.5,183)	164	6.253	15.895	12.271	false	false	false	false	false	**true**
do(startMove(cpto,loc(11.11,4.87),loc(9.9,5.66),184)	165	6.253	15.895	12.271	false	false	false	false	**true**	false
do(endMove(cpto,loc(11.11,4.87),loc(9.9,5.66),185)	166	6.631	17.009	19.523	false	false	false	false	false	**true**
do(startPan(7.5,186)	167	6.631	17.009	19.523	false	false	false	false	false	**true**
do(endPan(7.5,187)	168	6.631	17.009	19.523	false	false	false	false	false	**true**
do(startMove(cpto,loc(9.9,5.66),loc(8.52,6.35),188)	169	6.631	17.009	19.523	false	false	false	false	**true**	false
do(endMove(cpto,loc(9.9,5.66),loc(8.52,6.35),189)	170	7.114	17.967	28.460	false	false	false	false	false	**true**
do(startPan(7.5,190)	171	7.114	17.967	28.460	false	false	false	false	false	**true**
do(endPan(7.5,191)	172	7.114	17.967	28.460	false	false	false	false	false	**true**
do(startMove(cpto,loc(8.52,6.35),loc(7.0,6.93),192)	173	7.114	17.967	28.460	false	false	false	false	**true**	false
do(endMove(cpto,loc(8.52,6.35),loc(7.0,6.93),193)	174	7.726	18.389	38.905	false	false	false	false	false	**true**
do(startPan(7.5,194)	175	7.726	18.389	38.905	false	false	false	false	false	**true**
do(endPan(7.5,195)	176	7.726	18.389	38.905	false	false	false	false	false	**true**
do(startMove(cpto,loc(7.0,6.93),loc(5.36,7.39),196)	177	7.726	18.389	38.905	false	false	false	false	**true**	false
do(endMove(cpto,loc(7.0,6.93),loc(5.36,7.39),197)	178	8.506	17.978	49.972	false	false	false	false	false	**true**
do(startPan(7.5,198)	179	8.506	17.978	49.972	false	false	false	false	false	**true**
do(endPan(7.5,199)	180	8.506	17.978	49.972	false	false	false	false	false	**true**
do(startMove(cpto,loc(5.36,7.39),loc(3.62,7.73),200)	181	8.506	17.978	49.972	false	false	false	false	**true**	false
do(endMove(cpto,loc(5.36,7.39),loc(3.62,7.73),201)	182	9.507	16.739	60.116	false	false	false	false	false	**true**
do(startPan(7.5,202)	183	9.507	16.739	60.116	false	false	false	false	false	**true**
do(endPan(7.5,203)	184	9.507	16.739	60.116	false	false	false	false	false	**true**
do(startMove(cpto,loc(3.62,7.73),loc(1.83,7.93),204)	185	9.507	16.739	60.116	false	false	false	false	**true**	false
do(endMove(cpto,loc(3.62,7.73),loc(1.83,7.93),205)	186	10.789	15.081	67.762	false	false	false	false	false	**true**
do(startPan(7.5,206)	187	10.789	15.081	67.762	false	false	false	false	false	**true**
do(endPan(7.5,207)	188	10.789	15.081	67.762	false	false	false	false	false	**true**

Tabela 3 – Resultados dos testes de transição para pares de picos

(conclusão)

action	#	size A	size B	dist	approaching	coalescing	hiding	splitting	receding	two_peak_static
do(startMove(cpto,loc(1.83,7.93),loc(0.0,8.0),208)	189	10.789	15.081	67.762	false	false	false	false	**true**	false
do(endMove(cpto,loc(1.83,7.93),loc(0.0,8.0),209)	190	12.454	13.351	72.034	false	false	false	false	false	**true**
do(startPan(7.5,210)	191	12.454	13.351	72.034	false	false	false	false	false	**true**
do(endPan(7.5,211)	192	12.454	13.351	72.034	false	false	false	false	false	**true**

Fonte: Autor.