

CENTRO UNIVERSITÁRIO DA FEI

ADELINE CECILIA SÉCOLO

**RACIOCÍNIO ESPACIAL QUALITATIVO PARA
UM SISTEMA DE SEGURANÇA EM TRÁFEGO
DE VEÍCULOS**

ADELINE CECILIA SÉCOLO

**RACIOCÍNIO ESPACIAL QUALITATIVO PARA
UM SISTEMA DE SEGURANÇA EM TRÁFEGO
DE VEÍCULOS**

Dissertação de Mestrado apresentada ao Centro
Universitário da FEI para obtenção do título de
Mestre em Engenharia Elétrica, orientada pelo(a)
Prof. Dr. Paulo Eduardo Santos.

Sécolo, Adeline Cecilia

Raciocínio espacial qualitativo para um sistema de segurança em tráfego de veículos / Adeline Cecilia Sécolo. São Bernardo do Campo, 2011.

74 f. : il.

Dissertação de Mestrado - Centro Universitário da FEI.

Orientador: Prof. Dr. Paulo Eduardo Santos

1. Raciocínio Espacial Qualitativo. 2. Veículo Inteligente. I. Santos, Paulo Eduardo, orient. II. Título.

CDU 007.5



Centro Universitário da **FEI**

APRESENTAÇÃO DE DISSERTAÇÃO ATA DA BANCA JULGADORA

PGE- 10

Programa de Mestrado de Engenharia Elétrica

Aluna: Adeline Cecília Sécolo

Matrícula: 1081058

Título do Trabalho: **RACIOCÍNIO ESPACIAL QUALITATIVO PARA UM SISTEMA DE SEGURANÇA EM TRÁFEGO DE VEÍCULOS.**

Área de Concentração: Inteligência Artificial Aplicada à Automação

Orientador: Prof. Dr. Paulo Eduardo Santos

Data da realização da defesa: 22 / junho / 2011

ORIGINAL ASSINADA

A Banca Julgadora abaixo-assinada atribuiu ao candidato o seguinte:

APROVADO ☒

REPROVADO ☐

São Bernardo do Campo, 22 / junho / 2011.

MEMBROS DA BANCA JULGADORA

Prof. Dr. Paulo Eduardo Santos

Ass.: _____

Prof. Dr. Flavio Tonidandel

Ass.: _____

Prof. Dr. Flavio Soares Correa da Silva

Ass.: _____

VERSÃO FINAL DA DISSERTAÇÃO

**ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS
RECOMENDAÇÕES DA BANCA EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Carlos Eduardo Thomaz

Aos meus pais e marido. A todos que me incentivaram e não me deixaram desistir, acreditando em mim mais do que eu mesma.

AGRADECIMENTOS

Primeiramente a Deus, pela oportunidade de estudar mesmo vivendo em um país onde tantos nem mesmo chegam à graduação. Ao meu orientador Paulo Eduardo Santos, principalmente pela paciência. Aos professores de mestrado, pelo conhecimento transmitido, pelas dicas, por me consolarem e ajudarem quando perdi parte do meu texto da qualificação.

Aos colegas do mestrado por todas as tardes de finais de semana que passamos estudando e aprendendo juntos. À amiga Valquíria, que me ajudou com sua experiência de já ter passado por tudo isso antes.

A minha família, meus pais e principalmente ao meu marido, por compreenderem minha ausência em tantos eventos e pelo apoio e incentivo para continuar.

RESUMO

Um sistema de segurança em tráfego de veículos tem como objetivo dar suporte a um veículo inteligente para que este possa se locomover de forma segura entre outros veículos. Para isso, neste trabalho foram desenvolvidos axiomas que levam em consideração os dados adquiridos por um simulador de tráfego de veículos. Destes dados podem ser extraídas informações como distância entre o agente e os demais veículos, largura dos veículos, se os veículos estão se aproximando ou distanciando do agente, entre outras. Estes mesmos dados podem ser adquiridos através de um sistema de visão estéreo em um veículo inteligente. Em cada situação onde identificou-se potencial risco para o veículo inteligente, foi desenvolvida uma formalização qualitativa para identificá-lo. Desta maneira é possível sugerir ações ao agente, como por exemplo uma frenagem em tempo hábil para evitar uma colisão. Para os possíveis riscos formalizados foram considerados apenas veículos comuns, sendo assim, a interação com pedestres, animais ou objetos quaisquer não foi estudada neste trabalho. O sistema é capaz de interpretar o comportamento do veículo inteligente e dos veículos comuns e sugerir ações com o objetivo de manter o agente seguro, mas não faz com que o veículo inteligente mova-se de acordo com a ação sugerida. Também não está no escopo a implementação do sistema em um veículo real, ficando a proposta para trabalhos futuros.

Palavras-chave: Veículo Inteligente. Raciocínio Espacial Qualitativo.

ABSTRACT

The goal of a safety system for traffic vehicles is to help an intelligent vehicle so that it can move safely among other vehicles. In this work we developed some axioms that consider the data generated by a simulator of situations from vehicular traffic. These data give information about the environment, as the distance between the agent and other vehicles, the width of the vehicles, if the vehicles are getting closer or further from the agent. The same data can be obtained by a stereo camera in an intelligent vehicle. Each situation where a potential risk for the agent was found, a qualitative formalization was developed to identify this risk. It allows the system to suggest actions to the agent, as a braking needed on time to avoid a collision. For the possible risks formalized it was considered only common vehicles, so, the integration of pedestrians, animals or any other object was not studied in this work. The system is able to interpret the behavior of the agent and of the common vehicles and advice which actions the agent has to take to be safe, but do not make the agent to move according the action advised. The implementation on a real vehicle is not included in this work, this is left for future work.

Keywords: Intelligent Vehicles. Qualitative Spatial Reasoning.

LISTA DE TABELAS

6.1	Tabela do total de situações simuladas analisadas no experimento	46
-----	--	----

LISTA DE ILUSTRAÇÕES

1.1	Diagrama das fases de desenvolvimento do trabalho.	2
2.1	Diagrama de vizinhança do RCC	6
2.2	Tabela de relações do ROC - 20	8
3.1	Transições das Relações	16
3.2	Esquerda: visão global de uma cena com 2 objetos A e B e um observador v . Direita: perfil de profundidade relativo ao observador v na cena à esquerda.	20
3.3	Transição das relações	22
3.4	Diagrama de Vizinhança	24
4.1	Cenário e objetos	29
4.2	Dados mostrados durante a simulação	30
4.3	Vista global do cenário	31
4.4	Gráfico de Perfil de Profundidade	32
4.5	Mensagens de alerta	32
4.6	Áreas levadas em consideração na geração dos alertas	33
6.1	Cenário montado com agente e dois veículos comuns para exemplo.	44
6.2	Imagem dos picos que representam os veículos 1 e 2 em determinado instante da simulação.	44
6.3	Cena gerada durante a simulação de uma situação onde o sistema alerta a necessi- dade de parada do veículo inteligente.	47
6.4	Distância de segurança necessária para o sistema habilitar o agente a reiniciar movimento após uma parada.	48
6.5	Distância segura para ultrapassagem ou desvio de obstáculo pela faixa da direita.	48
6.6	Distância segura para ultrapassagem ou desvio de obstáculo pela faixa da esquerda.	49
6.7	A cena mostra que os veículos estão próximos sendo conveniente que o agente comece a desacelerar.	49

6.8	A distância entre os veículos ainda não demonstra grandes riscos mas considera-se prudente a partir deste ponto manter a velocidade constante.	50
6.9	A distância entre os veículos é grande o suficiente para permitir que o veículo inteligente acelere.	50
6.10	Picos representando o distanciamento entre o veículo inteligente e um veículo comum.	51

SUMÁRIO

1	INTRODUÇÃO	1
2	REVISÃO BIBLIOGRÁFICA	4
2.1	Raciocínio Espacial Qualitativo - REQ	4
2.1.1	Cálculo de Conexão de Regiões (<i>Region Conection Calculus - RCC</i>)	5
2.1.2	Cálculo de Oclusão de Regiões (<i>Region Occlusion Calculus - ROC</i>)	7
2.2	Cálculo de Situações (<i>Situation Calculus</i>)	10
2.3	Veículo Autônomo	13
3	TEORIA	15
3.1	Transições de Relações Entre Objetos	15
3.2	Cálculo de Perfil de Profundidade (<i>Depth Profile Calculus - DPC</i>)	19
3.3	Cálculo de Perfil de Profundidade Dinâmico (<i>Dynamic Depht Profile Calculus - DDPC</i>)	22
3.4	Raciocínio Sobre Perfil de Profundidade Dinâmico	25
4	SIMULADOR DE TRÁFEGO DE VEÍCULOS	28
4.1	Objetivo	28
4.2	O Sistema	28
4.3	Simulação	29
4.4	Interpretação dos Dados	30
4.5	Geração das Mensagens de Alerta	31
5	FORMALIZAÇÃO	34
5.1	Ações	34
5.2	Axiomas	35
5.3	Implementação	37

5.4	Regras em PROLOG	39
6	Testes	43
7	CONCLUSÃO	52
	Apêndice A – ARTIGOS PUBLICADOS	54
	REFERÊNCIAS	55

1 INTRODUÇÃO

Os problemas de tráfego de veículos reais existentes atualmente são inúmeros e muitos deles poderiam ser minimizados caso o veículo possuísse um sistema de raciocínio inteligente. Em diversas situações o auxílio da inteligência artificial poderia evitar acidentes e auxiliar o motorista a melhorar sua qualidade de vida no trânsito.

Para que isso seja possível no futuro, a cada ano desenvolvem-se trabalhos na área de pesquisa de veículos autônomos e assuntos relacionados, tais como o sistema apresentado em (SANTOS et al., 2009), onde um formalismo especifica o conteúdo geométrico de uma cena bem como as mudanças que acontecem nesta geometria. O trabalho descrito em (SANTOS et al., 2009) também propõe um algoritmo para construir a descrição destas mudanças a partir de deduções lógicas, interpretando desta maneira os movimentos dos objetos. Em (HUMMEL, 2009) foi feito o desenvolvimento de um sistema para identificação de interseções em vias de tráfego de veículos através de conjuntos de axiomas padrão.

No presente trabalho, a contribuição foi o desenvolvimento de uma teoria de raciocínio espacial qualitativo para situações de tráfego visando aplicações reais. Para isso, cada situação de risco foi estudada gerando uma formalização qualitativa para identificá-la. A aplicabilidade desta teoria foi testada em um ambiente simulado. O diagrama da figura 1.1 representa cada fase de desenvolvimento do trabalho.

Primeiramente observou-se o tráfego de veículos comuns, nas ruas da cidade e em vídeos, e também o tráfego de veículos autônomos em vídeos na internet. O objetivo destas observações foi identificar quais ações os motoristas ou sistemas de controle praticam para deslocar-se de um ponto a outro em segurança.

Definiu-se então as sete ações de segurança que seriam propostas neste trabalho. Foram descartadas as ações relacionadas ao percurso, como virar à direita ou à esquerda ou qualquer outra ação que tenha como objetivo apenas atingir o destino. Considerando-se assim apenas as ações parar, reiniciar movimento, mudar para faixa da direita, mudar para faixa da esquerda, acelerar, desacelerar e manter velocidade constante.

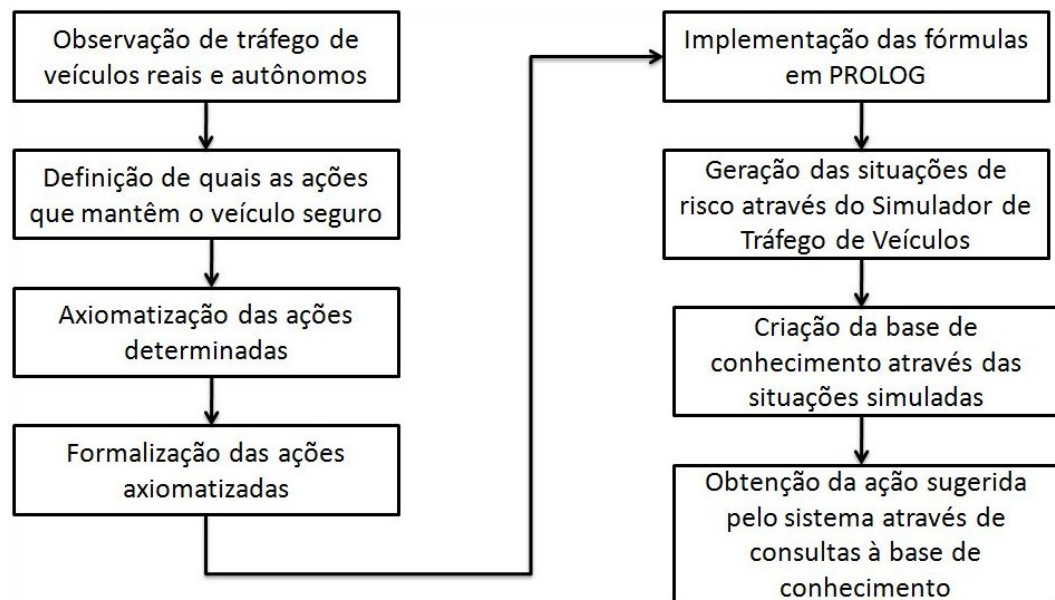


Figura 1.1: Diagrama das fases de desenvolvimento do trabalho.

As ações definidas foram axiomatizadas e em seguida foram desenvolvidas as fórmulas que representam tais axiomas. Estas fórmulas foram implementadas em PROLOG com o objetivo de testar para cada situação de risco, qual a ação sugerida pelo sistema através de consultas ao PROLOG.

As consultas ao PROLOG são feitas à uma base de dados, que neste caso foi gerada pelo Simulador de Tráfego de Veículos (capítulo 4). Diversos cenários contendo o veículo autônomo e veículos comuns foram simulados, gerando dados para a base de conhecimento. Para cada situação simulada foi gerada uma base de conhecimento que reflete os dados da cena. As consultas foram feitas a estas bases de conhecimento, obtendo como resposta do sistema qual a ação cabível naquele instante para manter o veículo autônomo seguro para seguir adiante até o destino.

Para o desenvolvimento deste projeto foram utilizados conceitos baseados na área de Raciocínio Espacial Qualitativo em Inteligência Artificial (seção 2.1), onde busca-se formalizar o conhecimento espacial, possibilitando a interpretação do comportamento de um sistema físico. Em particular utilizou-se na formação dos axiomas que descrevem as situações encontradas no tráfego de veículos o Cálculo de Perfil de Profundidade (seção 3.2), de onde extraem-se atributos como tamanho, distância e profundidade de objetos presentes em uma sequência de imagens, e o Cálculo de Situações (seção 2.2), que modela o comportamento de sistemas dinâmicos.

O Cálculo de Perfil de Profundidade Dinâmico (seção 3.3) é uma extensão do Cálculo de Perfil de Profundidade (seção 3.2), onde as alterações nas cenas podem gerar interpretações

dos movimentos dos objetos e do observador. Este formalismo foi utilizado neste trabalho na identificação de atributos como tamanho, profundidade e distância de objetos, assim como suas transições. Com estes atributos foi gerado o mapa de profundidade, utilizado como primitiva para o sistema de raciocínio qualitativo (capítulo 5).

Testes utilizando este formalismo em um ambiente simulado demonstrou bom desempenho na tarefa de interpretação de situações típicas em tráfego de veículos. Estes testes são descritos no capítulo 6.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão abordados os assuntos estudados como base para o desenvolvimento do trabalho desta dissertação, que consiste em desenvolver um sistema capaz de interpretar e inferir o comportamento de veículos a partir do ponto de vista de um veículo inteligente. Faremos uma revisão bibliográfica sobre Raciocínio Espacial Qualitativo, Carro Autônomo e Raciocínio de Ações.

2.1 Raciocínio Espacial Qualitativo - REQ

O Raciocínio Espacial Qualitativo (REQ) é uma área da Inteligência Artificial que busca formalizar e inferir fatos sobre a estrutura espacial de um domínio sem a utilização das tradicionais técnicas quantitativas. Além disso, visa tornar explícito o conhecimento do senso comum utilizado no dia a dia associado às abstrações utilizadas para criar modelos matemáticos (COHN et al., 1997). O REQ utiliza o conhecimento espacial, por exemplo, através de modelos que incluem topologia, direção, forma e distância entre objetos. Há diversas representações para estes atributos, e suas transições são características chave para o raciocínio no espaço qualitativo (COHN; HAZARIKA, 2001).

A partir de atributos a respeito da topologia, direção, forma, distância e suas transições criaram-se diversas modelagens com a utilização de axiomas a respeito do conhecimento sobre o espaço.

Em um sistema de REQ, corpos podem ser definidos como objetos físicos, enquanto regiões podem ser divididas em dois conjuntos: tridimensionais, que denotam o volume ocupado pelo corpo, e bidimensionais, que denotam a imagem do corpo projetada, visualizada de um ponto de vista (RANDELL; WITKOWSKI; SHANAHAN, 2001).

Dentre as diversas aplicações do raciocínio espacial qualitativo, daremos ênfase nas utilizadas em pesquisas relacionadas a veículos autônomos. No trabalho descrito em (HUMMEL, 2009) uma formalização espacial é utilizada na identificação de intersecções em vias urbanas.

Já em (COHN et al., 2006) há uma integração entre representações quantitativas e qualitativas no aprendizado de um agente. É comum a utilização do REQ em conjunto com outros métodos para obtenção de informações, como dados de sensores ou câmeras e análise estatística de dados (COHN; COHN; HOGG, 1998).

No trabalho (SANTOS, 2003) foi desenvolvido um sistema de interpretação de dados de sensores em um ambiente. Ainda em ambientes dinâmicos, os artigos (YAMAN; NAU; SUBRAHMANYAN, 2004) e (MULLER, 1998) abordam a utilização do REQ no raciocínio sobre objetos através da formalização de uma lógica de movimento (Logic of Motion - LOM). As aplicações desta lógica vão desde o monitoramento em tempo real de diversas aeronaves até a obtenção de informações sobre o deslocamento de pessoas com celulares, para melhor escolha do posicionamento de torres de transmissão, por exemplo. Esta é uma das poucas teorias de Raciocínio Espacial Qualitativo utilizada em veículos autônomos.

Dentre os atributos do REQ, a característica região foi utilizada para o desenvolvimento do cálculo de conexão de regiões, que terá sua teoria abordada adiante.

2.1.1 Cálculo de Conexão de Regiões (*Region Connection Calculus - RCC*)

O Cálculo de Conexão de Regiões (RCC) (RANDELL; CUI; COHN, 1992) consiste em um modelo de representação e raciocínio sobre relações a respeito de regiões espaciais. A base destas relações é a primitiva binária “conectado” $C(a,b)$, lida como “a região a está conectada à região b ”. Esta primitiva é verdadeira se as regiões a e b possuem ao menos um ponto em comum.

As relações definidas pelo RCC são:

- a) $DC(a,b)$ - a está desconectada de b ;
- b) $P(a,b)$ - a é parte de b ;
- c) $PP(a,b)$ - a é parte própria de b ;
- d) $EQ(a,b)$ - a é idêntica a b ;
- e) $O(a,b)$ - a sobrepõe b ;
- f) $DR(a,b)$ - a é discreta de b ;
- g) $PO(a,b)$ - a sobrepõe parcialmente b ;

- h) $EC(a, b)$ - a está externamente conectada a b ;
- i) $TPP(a, b)$ - a é parte própria tangencial de b ;
- j) $NTPP(a, b)$ - a é parte própria não tangencial de b .

A maioria das relações do RCC são simétricas, exceto TPP e $NTPP$, que por sua vez comportam relações inversas TPP^{-1} e $NTPP^{-1}$. As relações do RCC são mutuamente exclusivas, ou seja, duas regiões não podem ser representadas por mais de uma relação do RCC (RANDELL; CUI; COHN, 1992). A figura 2.1 apresenta um subconjunto do RCC e seu diagrama de vizinhança, em que cada aresta representa uma transição contínua de uma relação à outra. Isto é, entre duas relações conectadas por uma aresta não há qualquer outra relação possível.

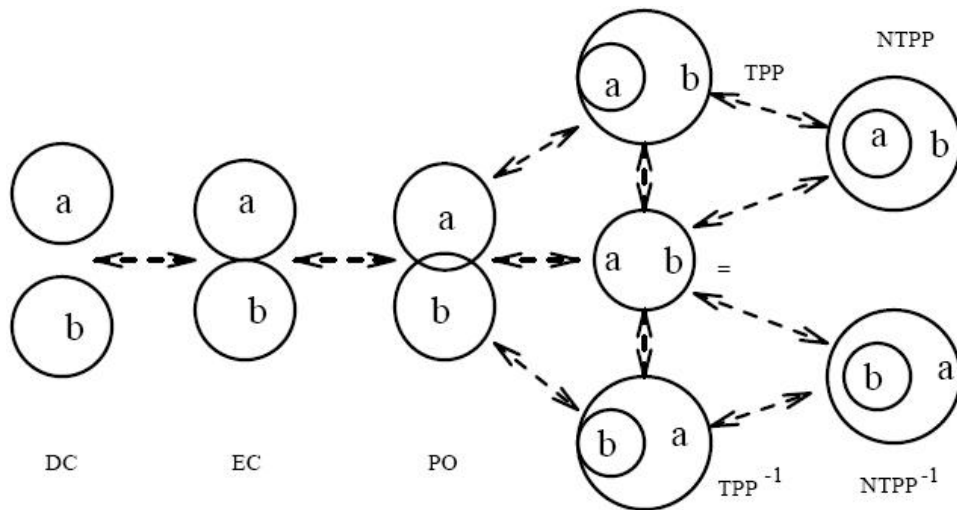


Figura 2.1: Diagrama de vizinhança do RCC
Fonte: (RANDELL; CUI; COHN, 1992)

Os axiomas das relações RCC são:

$$(A1) \forall x C(x, x)$$

$$(A2) \forall x \forall y (C(x, y) \rightarrow C(y, x))$$

As definições (D) das relações RCC são:

$$(D1) DC(x, y) \equiv_{def} \neg C(x, y)$$

$$(D2) P(x, y) \equiv_{def} \forall z (C(z, x) \rightarrow C(z, y))$$

$$(D3) EQ(x, y) \equiv_{def} P(x, y) \wedge P(y, x)$$

$$(D4) O(x, y) \equiv_{def} \exists z (P(z, x) \wedge P(z, y))$$

$$(D5) DR(x, y) \equiv_{def} \neg O(x, y)$$

$$(D6) PO(x, y) \equiv_{def} O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$$

$$(D7) EC(x, y) \equiv_{def} C(x, y) \wedge \neg O(x, y)$$

$$(D8) PP(x, y) \equiv_{def} P(x, y) \wedge \neg P(y, x)$$

$$(D9) TPP(x, y) \equiv_{def} PP(x, y) \wedge \exists z (EC(z, x) \wedge EC(z, y))$$

$$(D10) NTPP(x, y) \equiv_{def} PP(x, y) \wedge \neg \exists z (EC(z, x) \wedge EC(z, y))$$

$$(D11) PI(x, y) \equiv_{def} P(y, x)$$

$$(D12) PPI(x, y) \equiv_{def} PP(y, x)$$

$$(D13) TPPI(x, y) \equiv_{def} TPP(y, x)$$

$$(D14) NTPPI(x, y) \equiv_{def} NTPP(y, x)$$

A partir destas relações, o RCC é estendido ao Cálculo de Oclusão de Regiões (RANDELL; WITKOWSKI; SHANAHAN, 2001), explicado adiante.

No presente trabalho utilizamos um formalismo ou REQ (descrito no capítulo 3) que foi baseado no RCC e no cálculo de oclusão de regiões.

2.1.2 Cálculo de Oclusão de Regiões (*Region Occlusion Calculus - ROC*)

O Cálculo de Oclusão de Regiões - ROC (RANDELL; WITKOWSKI; SHANAHAN, 2001) utiliza as relações do RCC para modelar a interposição entre corpos, volumes e imagens. Esta teoria descreve 20 relações básicas de alinhamentos possíveis entre dois objetos, conforme figura 2.2.

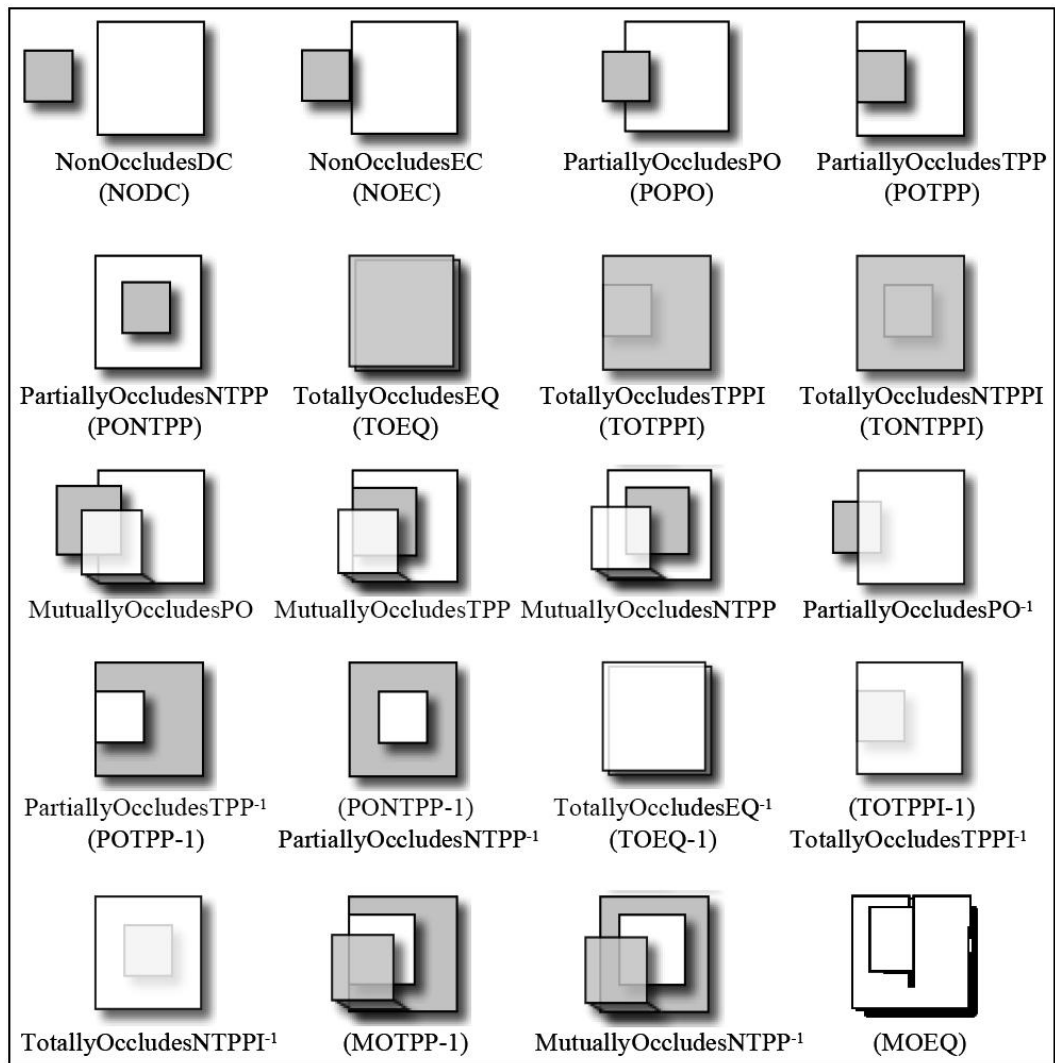


Figura 2.2: Tabela de relações do ROC - 20

Fonte: (RANDELL; WITKOWSKI, 2002)

ROC utiliza duas funções para diferenciar corpos e regiões: $region(x)$, que representa “a região ocupada por x ” e $image(x, v)$, que representa “a imagem de x em relação a um ponto de vista v ”. A função $region$ refere-se ao volume no espaço ocupado pelo corpo, enquanto a função $image$ refere-se à imagem projetada pelo corpo em um plano bidimensional em relação a um ponto de vista (v).

A relação primitiva que define oclusão em (RANDELL; WITKOWSKI; SHANAHAN, 2001) é:

$TotallyOccludes(x, y, v)$, que representa “ x oclui totalmente y do ponto de vista v ”.

Esta primitiva é irreflexiva e transitiva, e seus axiomas são:

$$(A3) \forall x \forall v \neg TotallyOccludes(x, x, v)$$

$$(A4) \forall x \forall y \forall z \forall v [[TotallyOccludes(x, y, v) \& TotallyOccludes(y, z, v)] \rightarrow TotallyOccludes(x, z, v)]$$

É possível definir outras relações de oclusão a partir desta primitiva e do RCC 2.1.1, tais como as seguintes:

- a) *PartiallyOccludes*(x, y, v), lida como “ x oclui parcialmente y do ponto de vista v ”;
- b) *MutuallyOccludes*(x, y, v), lida como “ x e y se ocluem mutuamente do ponto de vista v ”;
- c) *NonOccludes*(x, y, v), lida como “ x não oclui y do ponto de vista v ”.

Estas relações são associadas às relações $DC, EQ, PO, EC, TPP, NTPP, TPP^{-1}$ e $NTPP^{-1}$ e formam o conjunto de axiomas descritos abaixo, que representam as situações mostradas na figura 2.2.

As associações possíveis entre as relações primitivas e as relações do RCC seguem a seguinte formalização:

$$\Phi\Psi(x, y, v) \equiv \Phi(x, y, v) \wedge \Psi(image(x, v), image(y, v))$$

$$\Xi\Psi^{-1}(x, y, v) \equiv \Xi(y, x, v) \wedge \Psi(image(y, v), image(x, v))$$

Onde:

$$\Phi = NonOccludes, \text{ então } \Psi \in \{DC, EC\}$$

$$\Phi = TotallyOccludes, \text{ então } \Psi \in \{EQ, TTPI, NTPPI\}$$

$$\Phi = PartiallyOccludes, \text{ então } \Psi \in \{PO, TPP, NTPP\}$$

$$\Phi = MutuallyOccludes, \text{ então } \Psi \in \{PO, EQ, TPP, NTPP\}$$

em que:

$$\Xi = \textit{TotallyOccludes}, \text{ então } \Psi \in \{EQ, TPPI, NTPPI\}$$

$$\Xi = \textit{PartiallyOccludes}, \text{ então } \Psi \in \{PO, TPP, NTPP\}$$

$$\Xi = \textit{MutuallyOccludes}, \text{ então } \Psi \in \{TPP, NTPP\}$$

Tais relações são úteis na representação de conhecimentos extraídos de uma única figura ou uma sequência de figuras quando dados quantitativos não são necessários ou apropriados (KÖHLER; LUDWIGS, 2002). Estes aspectos são levados em consideração principalmente quando há objetos em movimento, como no trabalho apresentado em (FRANK et al., 1996), onde uma das dificuldades do rastreamento de veículos, a oclusão temporária, é abordada. Esta teoria também é utilizada em (SANTOS; FENELON; DEE, 2010) para a auto-localização qualitativa de um agente através da observação de um conjunto de sombras.

Um outro formalismo que permite inferir fatos em relação a ações e transições de estado, é a teoria descrita adiante, denominada Cálculo de Situações.

2.2 Cálculo de Situações (*Situation Calculus*)

O Cálculo de Situações (LEVESQUE; PIRRI; REITER, 1998) é um formalismo lógico utilizado para representação e raciocínio sobre ações e mudanças de estado em domínios dinâmicos. Utiliza-se de axiomas que consideram situações como histórias, onde cada situação é uma sequência finita de ações. A base dos axiomas é a função binária $s' = do(a, s)$, que representa a chegada à situação sucessora s' após a execução da ação a no estado s . O símbolo S_0 representa uma situação inicial. O predicado $Poss(a, s)$ é binário e representa a possibilidade de executar-se a ação a na situação s (pré-condição). A propriedade de Markov garante que o valor de uma situação sucessora é determinada unicamente pelas propriedades da situação atual, e não por qualquer outra situação.

As propriedades básicas das situações em qualquer domínio são compostas por quatro axiomas de domínio independentes para ações e fluentes, onde fluentes são descrições de situações que podem sofrer alterações com o tempo.

Os axiomas são:

(A9) $do(a_1, s_1) = do(a_2, s_2) \leftrightarrow a_1 = a_2 \wedge s_1 = s_2$, significa que duas situações são iguais se e somente se possuem a mesma sequência de ações. Uma vez que entende-se uma situação como uma história ou uma sequência de ações, duas situações são iguais se e somente se denotam histórias idênticas;

(A10) $\neg s \subset S_0$, significa que não há situação anterior à situação inicial;

(A11) $s \subset do(a, s') \equiv s \subseteq s'$, significa que a situação s' pode ser obtida da situação s adicionando-se uma ou mais ações à frente de s .

Estes axiomas, como todos os estudados neste capítulo, apresentam uma noção puramente qualitativa de tempo, onde é possível saber apenas se uma ação ocorre antes ou depois de outra, em uma situação.

Como exemplo de aplicação desta teoria, utilizaremos o robô que tem como função empilhar blocos. Para que o robô execute corretamente sua função, é necessário definir as ações primitivas, tais como: pegar objeto, soltar objeto, colocar objeto x sobre y e pegar o objeto x de cima de y .

Primeiro, vamos citar quais as ações primitivas:

- $primitive_action(pickup(x))$.
- $primitive_action(putdown(x))$.
- $primitive_action(stack(x, y))$.
- $primitive_action(unstack(x, y))$.

Em seguida, apresentamos as definições das ações primitivas:

- $pickup(x)$: pegar um bloco x , dado que não há nada sobre ele, está sobre a mesa e a garra do robô está livre.
- $putdown(x)$: colocar o bloco x sobre a mesa, dado que o robô o esteja segurando.
- $stack(x, y)$: colocar x sobre y , dado que o robô esteja segurando x e y esteja livre.

- $unstack(x,y)$: pegar x de sobre y , dado que a garra esteja livre, não há outro bloco sobre x e x está sobre y .

Fluentes devem ser definidos para descrever o estado do mundo:

- $clear(x,s)$: não há blocos sobre o bloco x na situação s .
- $on(x,y,s)$: bloco x está sobre o bloco y , na situação s .
- $ontable(x,s)$: bloco x está sobre a mesa, na situação s .
- $handempty(s)$: a garra do robô está vazia, na situação s .
- $holding(x,s)$: o robô está carregando o bloco x , na situação s .

Também define-se as pré-condições das ações primitivas, utilizando-se a função $poss(x,y)$:

- $poss(pickup(x),s) \leftarrow clear(x,s), ontable(x,s), handempty(s)$.
- $poss(putdown(x),s) \leftarrow holding(x,s)$.
- $poss(stack(x,y),s) \leftarrow holding(x,s), clear(y,s)$.
- $poss(unstack(x,y),s) \leftarrow on(x,y,s), handempty(s), clear(x,s)$.

Nesta etapa, os fluentes são formalizados:

- $clear(x,s) \leftarrow not(on(y,x,s))$.
- $on(x,y,s) \leftarrow not(clear(y,s)), not(holding(y,s))$.
- $ontable(x,s) \leftarrow holding(x,s), not(on(x,y,s))$.
- $handempty(s) \leftarrow not(holding(x,s)), not(holding(y,s))$.
- $holding(x,s) \leftarrow not(handempty(s)), not(on(y,x,s)), not(holding(y,s))$.

Para cada fluente, define-se um estado sucessor:

- $clear(x, do(a,s)) \leftarrow a = unstack(y,x); a = not(stack(x,y)), clear(x,s)$.
- $on(x,y, do(a,s)) \leftarrow a = stack(x,y); a = not(unstack(x,y)), on(x,y,s)$.

- $ontable(x, do(a, s)) \leftarrow a = putDown(x); not(pickup(x)), ontable(x, s).$
- $handempty(do(a, s)) \leftarrow a = putdown(x); a = stack(x, y); a = not(unstack(x, y)),$
 $handempty(s); a = not(pickup(x, y)), handempty(s).$
- $holding(x, do(a, s)) \leftarrow a = pickup(x); a = unstack(x, y); a = not(putdown(x)),$
 $holding(x, s); a = not(stack(x, y)), holding(x, s).$

Para quando nenhuma ação ocorreu ainda, define-se a situação inicial ($S0$).

- $ontable(bloco1, S0).$
- $ontable(bloco2, S0).$
- $clear(bloco1, S0).$
- $clear(bloco2, S0).$
- $handempty(S0).$

Desta maneira, axiomatizou-se o domínio do mundo dos blocos para um robô de uma garra.

No processo de axiomatização do sistema de segurança em tráfego de veículos a ser detalhado no capítulo 5 também foram definidas quais as ações possíveis, as regras que habilitam tais ações e em qual instante cada ação é sugerida pelo sistema.

2.3 Veículo Autônomo

Nesta seção serão apresentadas teorias voltadas à representação qualitativa e raciocínio sobre movimento de veículos, que é a sub-área onde o presente trabalho se insere.

Movimento é a chave para o entendimento das relações espaciais (MULLER, 1998). Conforme a teoria do RCC, descrita na seção 2.1.1, mudanças nos estados são analisadas de acordo com as transições dispostas no diagrama de vizinhança. Sendo assim, somente algumas transições são possíveis, assumindo-se a continuidade das mesmas.

Continuidade tem um papel importante no estudo sobre veículos autônomos, pois pode apresentar a trajetória de objetos físicos e eventos. Sendo assim, movimento é considerado uma

percepção de mudança contínua no espaço. No trabalho (MULLER, 1998), é apresentado um modelo no qual as relações primitivas são regiões espaço-temporais a partir das quais são definidas relações mais complexas sobre o movimento de veículos.

Para expressar alguns tipos de movimento, foi proposto um modelo denominado Lógica de Movimento (*Logic of Motion - LOM*) em (YAMAN; NAU; SUBRAHMANIAN, 2004). Este trabalho aborda questões sobre movimentos passados de veículos e projeções futuras. Os predicados *go*, *near* e *in* são utilizados para responder estas questões. Quando deseja-se saber se um objeto deslocou-se de sua posição inicial até uma outra posição qualquer em um intervalo de tempo t a uma velocidade v , utiliza-se o predicado *go*. Para determinar quais veículos estão alocados em uma determinada região retangular em um intervalo de tempo, utiliza-se o predicado *in*. Para saber a distância entre pares de objetos em um determinado ponto no tempo, utiliza-se o predicado *near*.

Complementando o trabalho descrito no parágrafo anterior, em (YAMAN; NAU; SUBRAHMANIAN, 2005) foi apresentado o algoritmo *CheckFar*, que possibilita saber se dois objetos estão suficientemente afastados um do outro durante um determinado intervalo de tempo.

Alguns cálculos em conjunto com a lógica expressam propriedades sobre o tempo de operação de sistemas móveis (MERZ; WIRSING; ZAPPE, 2003). Tais lógicas incluem modalidades espaciais e temporais para representar a estrutura topológica do sistema, bem como sua evolução no tempo. A identificação de um sistema móvel leva em consideração aspectos da topologia da rede, do comportamento do sistema dinâmico, além de movimentos e interação dos agentes.

No presente trabalho desenvolvemos axiomas que permitem a interação segura entre agente e objetos, considerando os dados obtidos de um ambiente dinâmico, tais como distância entre objeto e agente, largura dos objetos, se o objeto está se aproximando do agente, entre outros dados que foram utilizados na criação das regras que serão demonstradas no capítulo 5.

3 TEORIA

No presente capítulo descreveremos algumas teorias de REQ sobre transições entre objetos: o cálculo de perfil de profundidade e raciocínio sobre cálculo de perfil de profundidade dinâmico. Estes formalismos serão utilizados neste trabalho para construir um sistema de raciocínio automático para um carro inteligente em situações de tráfego de veículos.

3.1 Transições de Relações Entre Objetos

O trabalho descrito em (SANTOS; SHANAHAN, 2002) aplica as relações do RCC e ROC descritos no capítulo 2, para a caracterização dinâmica dos dados (que envolvem espaço e tempo) obtidos por uma câmera de visão estéreo localizada em um agente para representar as transições das relações espaciais de objetos. Há três conjuntos de axiomas em (SANTOS; SHANAHAN, 2002) que formam uma hierarquia para interpretação dos dados obtidos, a qual é dividida em três sub-tarefas.

Na primeira sub-tarefa, a imagem em um determinado momento é representada por regiões 2D. Uma linguagem similar à do RCC é utilizada pra representar as relações entre estas regiões. Nesta fase, a distância entre cada par de regiões é uma função primitiva onde é definido o grau de conectividade entre elas e pode ser extraída diretamente dos dados do sensor. Um dos termos desta linguagem é a função “distância” representada por $dist(x, y)$ e lida como a distância entre as regiões x e y . Esta função pode ser definida como o comprimento da menor linha que conecta quaisquer dois pontos nas duas bordas de regiões. Dentre as diversas relações entre regiões possíveis, são utilizados os predicados *Disc* (desconectado), *ExtC* (externamente conectado) e *Co* (conectado), devido à possibilidade de distingui-los analisando-se os dados do sensor. A axiomatização destas relações utiliza o símbolo δ_i como um valor pré-determinado de distância, tal qual representado nas fórmulas (A12) que significa que x está desconectado de y no tempo t se a distância entre eles for maior que δ_2 , (A13) onde observa-se que x está externamente conectado a y no tempo t se a distância entre eles for maior que δ_1 e menor que δ_2 e (A14) que atesta que x está sobrepondo y no tempo t se a distância entre eles for menor ou igual a δ_1 (SANTOS; SHANAHAN, 2002). Desta maneira, é possível classificar as imagens dos objetos no espaço em termos de relações de “distanciamento”, conforme observado na figura 3.1.

$$(A12) \quad DisC(x, y, t) \leftrightarrow (dist(x, y, t) > \delta_2);$$

$$(A13) \quad ExtC(x, y, t) \leftrightarrow (\delta_1 < dist(x, y, t)) \wedge (dist(x, y, t) \leq \delta_2);$$

$$(A14) \quad Co(x, y, t) \leftrightarrow dist(x, y, t) \leq \delta_1.$$

Os símbolos δ_1 e δ_2 são os valores limiares no domínio da distância. Para disparidade, o valor limiar é representado pelo símbolo μ e para tamanho o símbolo é σ .

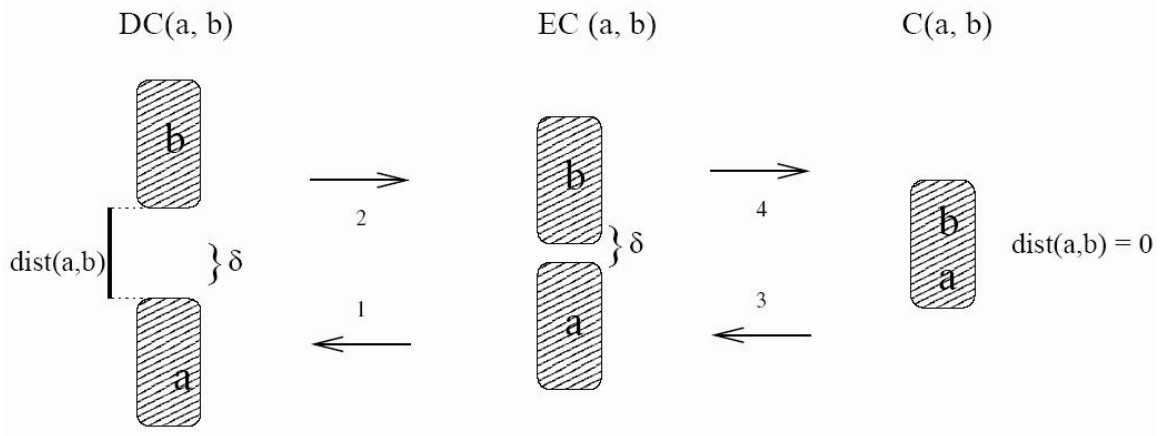


Figura 3.1: Transições das Relações
Fonte: (SANTOS; SHANAHAN, 2002)

Na segunda sub-tarefa são coletadas imagens instantâneas em ordem cronológica e representadas as transições das relações entre os objetos. Estas imagens são captadas pelo sensor do agente e contém informações completas, ou seja, incluem reflexos, oclusões e ruídos. Para representar puramente o mapeamento dos objetos visuais, pontos de vista e tempo para regiões espaciais de tais imagens, é utilizada a função $i(x, v, t)$, lida como “a imagem x , vista de v no tempo t ”. Assumindo-se que a e b sejam dois objetos visuais distintos, observados pelo sistema de visão do agente, as transições entre as relações de deslocamento entre eles são representadas pelos seguintes predicados:

- a) *approaching*($i(a, v, t), i(b, v, t)$) - lida como “as imagens de a e b estão se aproximando de acordo com o ponto de vista v no tempo t ”;
- b) *receding*($i(a, v, t), i(b, v, t)$) - lida como “as imagens de a e b estão se distanciando de acordo com o ponto de vista v no tempo t ”;
- c) *coalescing*($i(a, v, t), i(b, v, t)$) - lida como “as imagens de a e b estão se juntando de acordo com o ponto de vista v no tempo t ”;

d) *splitting*($i(a, v, t), i(b, v, t)$) - lida como “as imagens de a e b estão se dividindo de acordo com o ponto de vista v no tempo t ”.

Os axiomas para estes predicados (abaixo) assumem um intervalo de tempo $[t_1, t_2]$ curto o suficiente para excluir descontinuidades entre as fotos instantâneas do ambiente. Para ordenar os pontos de vista, utiliza-se o estado *between*(x, y, z), lido como “ x está entre y e z ”.

$$(A15) \text{ approaching}(i(a, v, t), i(b, v, t)) \leftarrow \exists t_1 t_2 v_1 v_2 (t_1 < t) \wedge (t < t_2) \wedge \text{between}(v, v_1, v_2) \wedge DC(i(a, v_1, t_1), i(b, v_1, t_1)) \wedge \neg Co(i(a, v_2, t_2), i(b, v_2, t_2)) \wedge (dist(i(a, v_1, t_1), i(b, v_1, t_1)) > dist(i(a, v_2, t_2), i(b, v_2, t_2))))$$

$$(A16) \text{ coalescing}(i(a, v, t), i(b, v, t)) \leftarrow \exists t_1 t_2 v_1 v_2 (t_1 < t) \wedge (t < t_2) \wedge \text{between}(v, v_1, v_2) \wedge (EC(i(a, v_1, t_1), i(b, v_1, t_1)) \vee DC(i(a, v_1, t_1), i(b, v_1, t_1))) \wedge Co(i(a, v_2, t_2), i(b, v_2, t_2)))$$

$$(A17) \text{ splitting}(i(a, v, t), i(b, v, t)) \leftarrow \exists t_1 t_2 v_1 v_2 (t_1 < t) \wedge (t < t_2) \wedge \text{between}(v, v_1, v_2) \wedge Co(i(a, v_1, t_1), i(b, v_1, t_1)) \wedge (EC(i(a, v_2, t_2), i(b, v_2, t_2)) \vee DC(i(a, v_2, t_2), i(b, v_2, t_2)))$$

$$(A18) \text{ receding}(i(a, v, t), i(b, v, t)) \leftarrow \exists t_1 t_2 v_1 v_2 (t_1 < t) \wedge (t < t_2) \wedge \text{between}(v, v_1, v_2) \wedge (EC(i(a, v_1, t_1), i(b, v_1, t_1)) \vee DC(i(a, v_1, t_1), i(b, v_1, t_1))) \wedge (dist(i(a, v_1, t_1), i(b, v_1, t_1)) < dist(i(a, v_2, t_2), i(b, v_2, t_2)))$$

$$(A19) \text{ static}(i(a, v, t), i(b, v, t)) \leftarrow \exists t_1 t_2 (t_1 < t) \wedge (t < t_2) \wedge (dist(i(a, v, t_1), i(b, v, t_1)) = dist(i(a, v, t_2), i(b, v, t_2)))$$

Os axiomas [A15] a [A19] são utilizados para explicar as diferenças entre pares de imagens consecutivas, classificando as transições dos dados do sensor e inferindo uma relação dinâmica entre as cinco possíveis.

As cinco relações apresentadas anteriormente foram reescritas em possíveis explicações para as transições percebidas através dos dados do sensor, resultando em sete novas relações objeto-observador possíveis. A terceira tarefa consiste em formar conjuntos de hipóteses sobre as relações espaciais dinâmicas entre os objetos que podem ter suas transições estimadas pelos dados do sensor.

a) *getting_closer*(a, b, v, t), é lida como “os objetos a e b estão se aproximando no tempo t e do ponto de vista v ”;

- b)** $ap_getting_closer(a, b, v, t)$, representa que “os objetos a e b estão aparentemente se aproximando no tempo t devido a movimentação do observador v ”;
- c)** $getting_further(a, b, v, t)$: “os objetos a e b estão se distanciando no tempo t e do ponto de vista v ”;
- d)** $ap_getting_further(a, b, v, t)$: “os objetos a e b estão aparentemente se distanciando no tempo t devido a movimentação do observador v ”;
- e)** $occluding(a, b, v, t)$: “um dos objetos a e b está se interpondo ao outro no tempo t do ponto de vista v ”;
- f)** $touching(a, b, v, t)$: “ a e b estão se tocando no tempo t conforme notado por v ”;
- g)** $static(a, b, v, t)$: “ a e b estão estáticos no tempo t ”.

O predicado $located(a, v, t)$, lido como “o corpo físico a está localizado em v no tempo t ”, é acrescentado para apresentar a conexão entre este novo conjunto de sete relações e as definições abstratas descritas. Pode-se definir a relação entre os predicados das imagens com os predicados dos corpos físicos, conforme os axiomas A20 ao A23 descritos a seguir.

$$(A20) \text{ approaching}(i(a, v, t), i(b, v, t)) \leftarrow located(robot, v, t) \wedge (getting_closer(a, b, v, t) \vee ap_getting_closer(a, b, v, t))$$

$$(A21) \text{ coalescing}(i(a, v, t), i(b, v, t)) \leftarrow located(robot, v, t) \wedge (occluding(a, b, v, t) \vee touching(a, b, v, t))$$

$$(A22) \text{ splitting}(i(a, v, t), i(b, v, t)) \vee receding((i(a, v, t), i(b, v, t))) \leftarrow located(robot, v, t) \wedge (getting_further(a, b, v, t) \vee ap_getting_further(a, b, v, t))$$

$$(A23) \text{ static}(i(a, v, t), i(b, v, t)) \leftarrow located(robot, v, t) \wedge static(a, b, v, t)$$

Após estas três fases (classificação dos deslocamentos, classificação das transições e possíveis explicações para tais transições), inicia-se o processo de explicação dos dados do sensor através de inferência e previsão das configurações futuras através de dedução. Os axiomas (A15) a (A19) são descrições das transições dos sensores uma vez que utilizam a função i (*imagem*), enquanto (A20) a (A23) são hipóteses sobre o que pode ter acontecido com os objetos a e b dadas as informações obtidas pelos sensores.

3.2 Cálculo de Perfil de Profundidade (*Depth Profile Calculus - DPC*)

Seguindo a mesma linha de raciocínio de (SANTOS; SHANAHAN, 2002), apresentado na seção 3.1, é proposta a teoria denominada Cálculo de Perfil de Profundidade (DPC - Depth Profile Calculus) (SANTOS, 2007). Esta teoria trata um modelamento que dá-se através de funções que extraem atributos como profundidade, tamanho e distância de objetos presentes em uma sequência de imagens instantâneas de um determinado ambiente. Os objetos são representados por picos de onde é possível identificar características como profundidade e movimento através do perfil de profundidade. As transições das relações destes atributos são representadas pelo Cálculo de Perfil de Profundidade Dinâmico (DDPC - Dynamic Depth Profile Calculus).

Obtendo-se dados de um sistema de visão estéreo é possível gerar hipóteses para obter conhecimento através de representação simbólica destes dados. O sistema provê um mapa de profundidade cuja simplificação é denominada linha horizontal, que cruza o centro do campo de visão. Destas linhas horizontais, extraem-se informações 2D de um corte plano de uma determinada cena, tais como tamanho e profundidade de objetos. A partir dos perfis de profundidade são gerados picos causados por objetos próximos ou conjuntos de objetos. Estes picos são considerados primitivas no sistema de raciocínio qualitativo. No lugar de profundidade, utiliza-se a informação de disparidade para representar a distância entre um objeto e o ponto de vista do observador, uma vez que esta informação pode ser obtida diretamente da câmera estéreo. Disparidade é inversamente proporcional à profundidade vezes uma constante dependente da geometria da câmera. Quanto maior o valor da disparidade, mais perto do observador está o objeto.

O perfil de profundidade é representado por um gráfico onde a medida de disparidade está no eixo y e os pixels da linha horizontal da cena estão no eixo x, conforme observa-se na figura 3.2. A letra L representa a maior distância entre o objeto e o observador que o sistema consegue identificar. O ponto cuja disparidade seja menor que L recebe valor menos um.

Além da disparidade, o perfil de profundidade apresenta informações sobre o tamanho relativo dos objetos e a distância entre eles do ponto de vista do observador. O valor do tamanho dos objetos é obtido através da diferença entre as duas bordas laterais do pico, enquanto a distância entre eles é obtida pela diferença entre as bordas laterais mais próximas de dois picos diferentes.

São assumidas as seguintes funções para representar picos, disparidade, tamanho e distância.

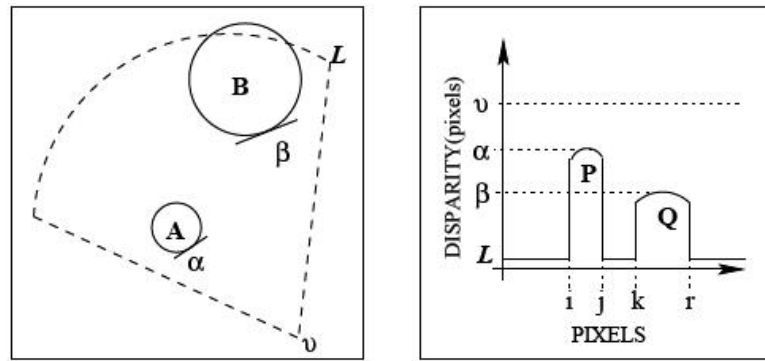


Figura 3.2: Esquerda: visão global de uma cena com 2 objetos A e B e um observador v .
 Direita: perfil de profundidade relativo ao observador v na cena à esquerda.
 Fonte: (SANTOS, 2007)

p_o : corpo físico x ponto no tempo $\rightarrow$ peak

$disp$: pico x ponto no tempo $\rightarrow$ disparidade

$size$: pico x pico x ponto no tempo $\rightarrow$ tamanho

$dist$: pico x pico x ponto no tempo $\rightarrow$ tamanho

Onde p_o pode ser lido como “ p é o pico do corpo b no tempo t ”, $disp$ dá a disparidade do pico em um determinado instante de tempo, $size$ mapeia um pico e um ponto no tempo para o tamanho do pico e $dist$ mapeia dois picos e um ponto no tempo para a distância angular que separa os dois picos neste instante. Por exemplo, na figura 3.2 (esq.) a distância ($dist$) entre os picos p e q é dada por $-j - k$ (i.e. é dada pela subtração entre as arestas verticais mais próximas entre dois picos).

Os principais predicados do sistema de raciocínio espacial para perfis de profundidade são as relações de distância, disparidade e tamanho. Para todas as relações $R(x,y,t)$, o primeiro argumento (x) refere-se ao pico situado à esquerda do pico referido no segundo argumento (y).

O cálculo de perfis de profundidade é formado por três conjuntos de relações (para distância, profundidade e tamanho) apresentadas logo adiante. Tais relações e suas transições podem ser visualizadas na figura 3.3.

As relações de distância são compostas pelos axiomas (A12), (A13) e (A14) citados na seção 3.1.

Um outro conjunto de axiomas trata das relações de disparidade conforme descrição a

seguir.

- a) $Further_Than(x, y, t) : FT(x, y, t)$ - que significa que x está mais longe do observador do que y ;
- b) $Closer_Than(x, y, t) : CT(x, y, t)$ - que significa que x está mais perto do observador do que y ;
- c) $Depth_Equal(x, y, t) : DEq(x, y, t)$ - que significa que x e y estão à mesma distância do observador.

Para estas relações, os axiomas são:

$$(A24) FT(x, y, t) \leftrightarrow (disp(x, t) < disp(y, t)) \wedge (|disp(x, t) - disp(y, t)| \geq \mu);$$

$$(A25) CT(x, y, t) \leftrightarrow (disp(x, t) > disp(y, t)) \wedge (|disp(x, t) - disp(y, t)| \geq \mu);$$

$$(A26) DEq(x, y, t) \leftrightarrow (|disp(x, t) - disp(y, t)| < \mu).$$

As relações sobre o atributo tamanho são definidas por:

- a) $Larger_Than(x, y, t) - LT(x, y, t)$ - que indica que x é maior do que y no tempo t ;
- b) $Smaller_Than(x, y, t) - ST(x, y, t)$ - que indica que x é menor do que y no tempo t ;
- c) $Size_Equal(x, y, t) - SEq(x, y, t)$ - que indica que x e y são do mesmo tamanho no tempo t .

E são restritas pelos axiomas [C1], [C2] E [C3] abaixo.

$$[C1] LT(x, y, t) \leftrightarrow (size(x, t) > size(y, t)) \wedge (|size(x, t) - size(y, t)| \geq \sigma);$$

$$[C2] ST(x, y, t) \leftrightarrow (size(x, t) < size(y, t)) \wedge (|size(x, t) - size(y, t)| \geq \sigma);$$

$$[C3] SEq(x, y, t) \leftrightarrow (|size(x, t) - size(y, t)| < \sigma).$$

No trabalho apresentado nesta dissertação, todas as informações contidas nos picos gerados pelo DPC serão utilizados para localização e locomoção do veículo inteligente. Até o momento

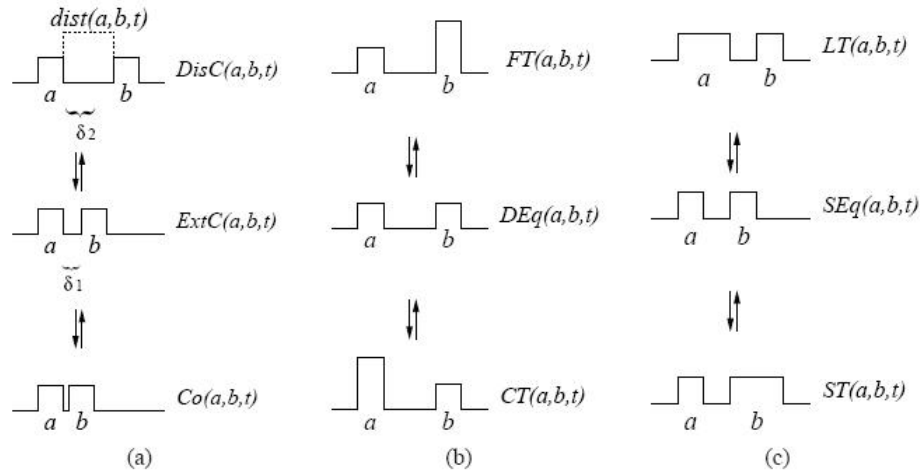


Figura 3.3: Transição das relações

Fonte: (SANTOS, 2007)

não há publicação de nenhum trabalho que utilize esta teoria como base, sendo nosso objetivo utilizar seus dados em uma aplicação real.

A seguir será apresentada a continuação desta teoria (DPC), onde os objetos estão em um ambiente dinâmico.

3.3 Cálculo de Perfil de Profundidade Dinâmico (*Dynamic Depht Profile Calculus - DDPC*)

O DDPC é uma extensão do DPC apresentado na seção 3.2. Seus predicados definem alterações nas cenas observadas gerando possíveis interpretações de movimentos dos objetos e do observador.

Neste domínio, assumem-se três condições básicas. A primeira assegura a persistência do objeto, onde o mesmo não pode simplesmente aparecer ou desaparecer. A segunda refere-se à continuidade, onde as alterações na localização dos objetos ocorre somente devido à movimentos contínuos e a terceira assegura que objetos não passam um através do outro, ou seja, são substanciais.

Novos axiomas são criados para a representação das transições. O predicado $In_sight(p, t)$ é verdadeiro se existe um objeto notado pelo observador, ou seja, $p = p_o(a, t)$ não é nulo. Um par de objetos é considerado dentro do campo de visão do observador se as relações definidas no DPC forem verdadeiras para dois picos distintos.

Os predicados sobre as transições dos picos são listados adiante.

- a) *Extending*(p, t), atesta que o pico está aumentando de largura no tempo t ;
- b) *Shrinking*(p, t), atesta que o pico está diminuindo de largura no tempo t ;
- c) *Vanishing*(p, t), atesta que o pico está diminuindo de tamanho no tempo t a ponto de não ser mais percebido;
- d) *Appearing*(p, t), atesta que um pico está surgindo do fundo de um perfil;
- e) *Peak_approaching_DisC*(p, q, t), representa dois picos se aproximando no tempo t mas permanecendo desconectados;
- f) *Peak_approaching_ExtC*(p, q, t), representa dois picos se aproximando no tempo t e tornando-se externamente conectados;
- g) *Peak_receding_DisC*(p, q, t), representa dois picos afastando-se um do outro no tempo t e permanecendo desconectados;
- h) *Peak_receding_ExtC*(p, q, t), representa dois picos externamente conectados afastando-se um do outro no tempo t ;
- i) *Peak_coalescing*(p, q, t), atesta que dois picos estão fundindo-se no tempo t ;
- j) *Going_out_sight*(p, q, t), representa o caso de um pico desaparecer do campo de visão do observador no tempo t (por oclusão ou por deslocamento lateral);
- k) *Peak_splitting*(p, q, t), representa o caso de um pico dividir-se em dois no tempo t ;
- l) *Single_peak_static*(q, t), atesta que o pico não sofreu alteração de disparidade ou tamanho no tempo t ;
- m) *Peak_static*(p, q, t), atesta que a distância entre dois pico não sofreu alteração no tempo t .

Estes predicados dão origem a axiomas similares aos da seção 3.1, que por sua vez expressam as transições dos atributos dos picos.

Para prever o comportamento futuro dos objetos, utiliza-se o diagrama de vizinhança das relações de transição do pico. É possível observar na figura 3.4 que nem todas as transições são reversíveis.

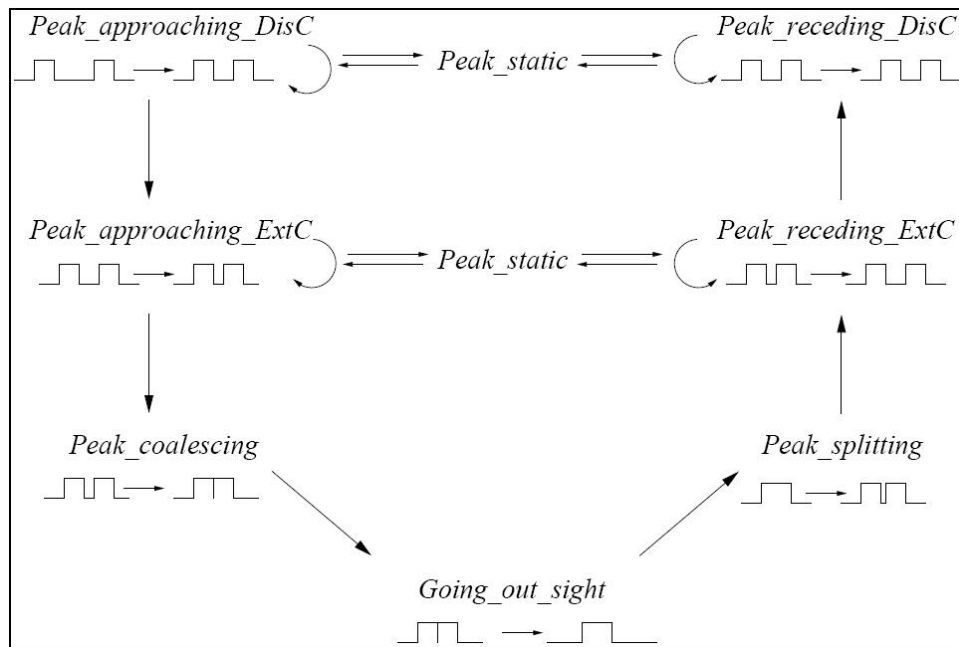


Figura 3.4: Diagrama de Vizinhança

Fonte: (SANTOS, 2007)

O processo de hipotetizar explicações sobre as observações acontece através da busca de diferenças entre pares consecutivos de imagens. Para tornar esta busca mais eficiente, é importante alimentar o sistema com informações sobre os objetos estáticos que fazem parte do ambiente, pois assim é possível inferir dados sobre movimentos do observador com relação a estes objetos.

Cada relação dos picos de profundidade está conectada a uma relação espacial dos corpos físicos do ponto de vista do observador e são chamadas hipóteses observador-objeto. Estas hipóteses são divididas em cinco classes:

- a) Extending, shrinking, vanishing e appearing;
- b) Approaching e receding;
- c) Coalescing e going out of sight;
- d) Splitting;
- e) Object/ peak static.

No trabalho (SANTOS et al., 2009) há mais um exemplo de aplicação das teorias envolvendo cenas dinâmicas. O sistema desenvolvido é capaz de interpretar com grande precisão movimentos

rotacionais de objetos de cores diferentes. Vale destacar que tal sistema analisa cenas continuamente, como num vídeo, enquanto outros fazem comparações entre pares de imagens.

Na próxima seção será descrito o raciocínio de ações aplicado ao perfil de profundidade dinâmico.

3.4 Raciocínio Sobre Perfil de Profundidade Dinâmico

O trabalho apresentado em (SOUTCHANSKI; SANTOS, 2008) é uma extensão do DDPC que engloba tanto raciocínio sobre dados obtidos por sensores como também sobre ações no ambiente estudado. Trata-se de um formalismo lógico que representa transições entre perfis de profundidade no Cálculo de Situações. Com o objetivo de manipular dados simbólicos sobre movimentos dos objetos e do próprio observador utiliza-se, além das relações espaciais qualitativas e cálculo de situações, uma descrição geométrica do ambiente. Por exemplo, é possível medir o deslocamento relativo de determinado objeto através da distância angular, que consiste simplesmente na diferença entre o ponto mais à esquerda de um perfil e o ponto mais à direita de outro. A distância Euclidiana é também utilizada, sendo representada pelo termo $euD(loc(x_1, y_1), loc(x_2, y_2), dist)$ sendo que $dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

O formalismo lógico é representado através de Axiomas de Estado Sucessor, ou seja, as transições entre os perfis de profundidade são executadas pelo Cálculo de Situações, onde cada transição identificada é causada por uma ação. Devido a este fato, são designados alguns termos para identificar tais ações. Uma linguagem tipada é utilizada para representar dados, tais como: perfis de profundidade (p), tempo (t), profundidade (u), tamanho (z), distância angular (d), ângulo de rotação (ω), direção da face do robô (θ), corpos físicos (b), coordenadas (x e y) e pontos de vista (v).

O movimento de um agente ou corpo é descrito pelo termo $startMove(b, l_1, l_2, time)$, lido como “ b inicia o movimento do ponto l_1 ao ponto l_2 no instante $time$ ”. O fim do movimento é descrito pelo termo $endMove$, similar e oposto ao termo anterior. Além disso, o agente pode girar horizontalmente, alterando a posição do cone de visão. Este movimento é descrito pelos termos $startPan(\omega, time)$ e $endPan(\omega, time)$, sendo ω positivo caso o movimento seja no sentido horário e negativo caso o movimento seja anti-horário. O termo $sense(p, loc(x_r, y_r), time)$ representa a ação para obtenção de um perfil p a partir de dados dos sensores.

Estas ações possuem pré-condições formais para serem executadas. Segue exemplo para a ação *startMove*, e as outras ações seguem a mesma linha de raciocínio.

$$poss(startMove(b, l_1, l_2, t), s) \equiv location(b, l_1, s) \wedge \neg \exists l, l' moving(b, l, l', s) \wedge l_1 \neq l_2 \wedge start(s, t') \wedge t \geq t'.$$

O termo $Poss(A(\vec{x}), s) \equiv \Pi_A(\vec{x}, s)$ significa que A é possível se e somente se a condição $\Pi_A(\vec{x}, s)$ for verdadeira.

Sempre que acontece alguma ação, nota-se uma alteração em algum atributo do pico identificado. Esta alteração enquadra-se em uma transição de um estado para outro, que pode ser em um único pico ou em um par deles.

As seguintes transições podem ser percebidas em um único pico: *extending*, *shrinking*, *appearing*, *vanishing* ou *static-peak* que acontece quando determinada ação não gera alteração nos atributos do pico.

Para um par de picos, temos as seguintes relações: *approaching*, *receding*, *coalescing*, *hiding*, *splitting* ou *two-peak-static* que representa dois picos sem transição alguma.

Para axiomatizar as pré-condições define-se a Teoria de Profundidade e Movimento (*Theory of Depth and Motion - TDM*). Em um domínio onde tanto os objetos, quanto o observador podem se mover, as percepções captadas pelos sensores em um intervalo de tempo podem ser representadas no diagrama de vizinhança (figura 3.4). Desta maneira é possível raciocinar sobre os movimentos dos objetos e do observador no ambiente e calcular seus efeitos através da geometria euclidiana. Para exemplificar o TDM, utilizaremos o pseudo-código da ação *approaching* a seguir.

$$approaching(peak1, peak2, viewpoint, do(a, s))$$

Esta ação será verdadeira se e somente se:

- a) *a é uma ação de medição do ângulo entre os picos e tal ação encontrou um ângulo menor do que era na situação s, ou;*
- b) *a é uma ação endMove, ou seja, é a interrupção de um movimento do robô já detectado anteriormente cujo resultado é a diminuição do ângulo entre os picos em relação à situação s, ou;*

- c) *a é uma ação endMove mas agora para terminar o movimento de um objeto, causando resultado similar ao descrito anteriormente, ou;*
- d) *após a percepção da ação approaching, a não representa qualquer ação que possa alterar esta condição, portanto, a percepção é mantida.*

O trabalho desta dissertação tem como objetivo utilizar o TDM para cercar todas as opções possíveis em um conceito de diagrama de vizinhança para os movimentos de um veículo inteligente. Além disso, pretende-se a implementação de tais axiomas em um simulador de tráfego dinâmico.

Apesar de não ficar explícito em (SOUTCHANSKI; SANTOS, 2008), pode-se considerar que após a percepção de determinada ação, se na próxima imagem o objeto não estiver mais no campo de visão do observador, é mantida como verdadeira a ação previamente observada. Porém, se o objeto ainda fizer parte do ponto de vista e nenhuma alteração em seus movimentos for notada, considera-se a situação estática (*static_peak*).

4 SIMULADOR DE TRÁFEGO DE VEÍCULOS

O simulador descrito neste capítulo será utilizado como base para a implementação das ações do carro inteligente. Atualmente, o sistema com o qual o simulador trabalha é capaz de identificar situações e sinalizar riscos, porém, nenhuma atitude é tomada.

4.1 Objetivo

O simulador foi inicialmente desenvolvido com o objetivo de ser capaz de gerar dados qualitativos e hipóteses sobre um ambiente bidimensional, onde apresentavam-se observador e objeto (FERRARI et al., 2008). Com alguns aperfeiçoamentos, chegou-se a um cenário onde o observador tornou-se um veículo inteligente, e os objetos tornaram-se veículos comuns e pedestres. Além disso, foram criadas ruas, como entidades estáticas. O trajeto de todos os veículos e pedestres presentes no ambiente simulado é definido pelo usuário e identificado através de pontos. Desta maneira, é possível simular situações de trânsito e estudar a tomada de decisões visando evitar acidentes (FERRARI et al., 2008).

4.2 O Sistema

A interface gráfica utilizada é renderizada por uma Interface de Programação de Aplicativos (API) Java 2D. o API é caracterizado por possuir conjuntos de pacotes que proveem as rotinas necessárias à criação e manipulação dos elementos gráficos. Utiliza-se a plataforma Eclipse para garantir a portabilidade para sistemas operacionais. O *workbench* é utilizado como interface gráfica com o usuário. Este recurso permite que o mesmo abra, edite, salve e organize editores e visões do cenário criado (FERRARI et al., 2008).

O usuário deve utilizar a interface gráfica para configurar o ambiente a ser simulado. Este ambiente é bidimensional sendo apresentado de um ponto de vista global. Existem três cenários diferentes de ruas, onde o usuário pode inserir o veículo observador, veículos não inteligentes e pedestres, bem como os respectivos percursos. Estes dados serão utilizados como entradas para o

cálculo dos movimentos.

4.3 Simulação

A simulação acontece em três etapas. Na primeira, define-se o ambiente conforme citado anteriormente e mostrado na figura 4.1. As linhas cinzas representam as vias, o retângulo amarelo é o observador enquanto o branco é o veículo comum. Cada situação a ser simulada pode conter apenas um veículo inteligente e até cinquenta veículos comuns. O ponto branco representa o pedestre e os pontos azuis são os percursos. Não é possível fazer uma simulação sem que o veículo inteligente seja inserido no cenário. Quanto aos objetos, foram realizados testes com a inserção de até cinquenta. O sistema funcionou bem, identificando os riscos, e o tempo de simulação aumentou linearmente ao número de objetos inseridos.

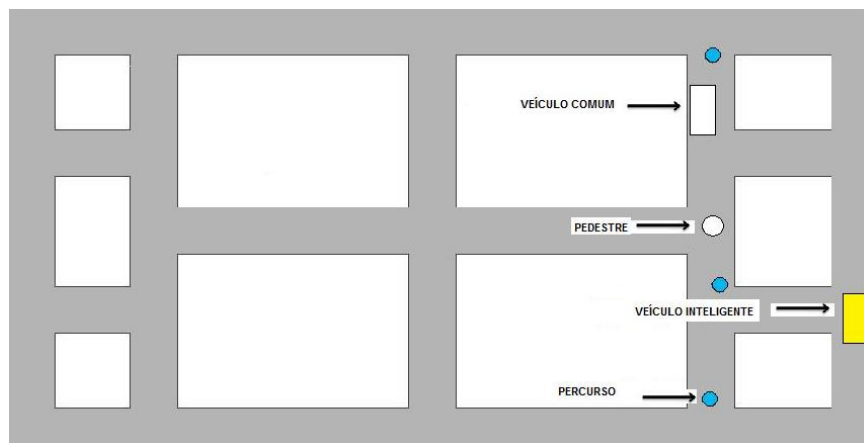


Figura 4.1: Cenário e objetos

Em seguida, com os objetos em movimento, gera-se o perfil de profundidade onde o observador é o veículo inteligente. Sua visão é linear e restrita pelo ângulo de abertura do cone de visão que é de 120° . Para gerar o perfil de profundidade são utilizadas relações trigonométricas entre o observador e os objetos. Quando o sistema detecta os objetos no campo de visão do observador, é gerada uma reta do ponto central de cada objeto ao ponto de origem do cone de visão e a partir daí obtém-se o coeficiente angular de cada objeto. Para obter a largura aproximada de cada objeto, considera-se que eles sejam circulares e calcula-se quantos graus da visão do observador o objeto ocupa. Todos estes dados formam um estado que é enviado ao Prolog para a geração das mensagens de alerta. Quando o sistema detecta um objeto, o mesmo é marcado fazendo com que, caso este objeto saia do cone de visão do agente e após alguns instantes volte a aparecer no cone

de visão, ele seja identificado com a mesma variável utilizada para descrevê-lo da primeira vez que foi detectado.

A cada iteração da simulação o sistema mostra três tipos de informações ao usuário, como pode ser visto na figura 4.2. Uma delas é o cenário com ponto de vista global (figura 4.3), mostrando os objetos em movimento até suas posições finais de acordo com os trajetos definidos. Também é mostrado o gráfico de perfil de profundidade (figura 4.4) e as mensagens de alerta (figura 4.5). Existem três mensagens de alerta implementadas no sistema, são elas: “Frenagem Recomendada”, “Risco de Colisão” e “Ultrapassagem Não Recomendada”. A cada par de gráficos de perfil de profundidade recebidos, o sistema infere relações espaciais dinâmicas entre observador e objetos. Com base nas relações inferidas, o sistema faz perguntas a uma base de dados do Prolog e cada resposta é mostrada ao usuário em forma de alertas.

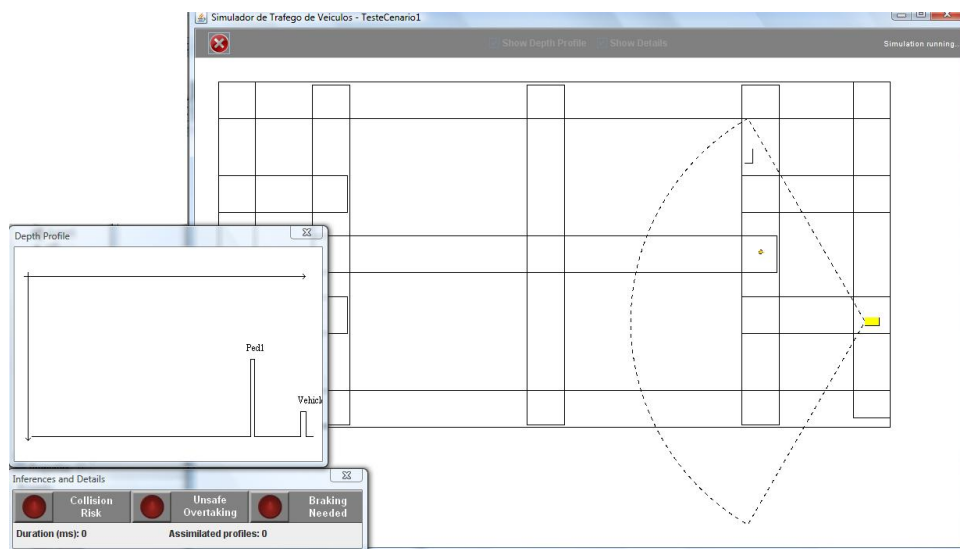


Figura 4.2: Dados mostrados durante a simulação

4.4 Interpretação dos Dados

A interpretação do perfil de profundidade é feita através da combinação dos dados de *background* e *foreground*. Quando o sensor do veículo detecta algum obstáculo entre ele e o fim de seu campo de visão, a informação é considerada *foreground* e possui a seguinte forma: *foreground* (ped1,30.3,3.65) onde é mostrado o nome do objeto detectado, distância objeto-observador,

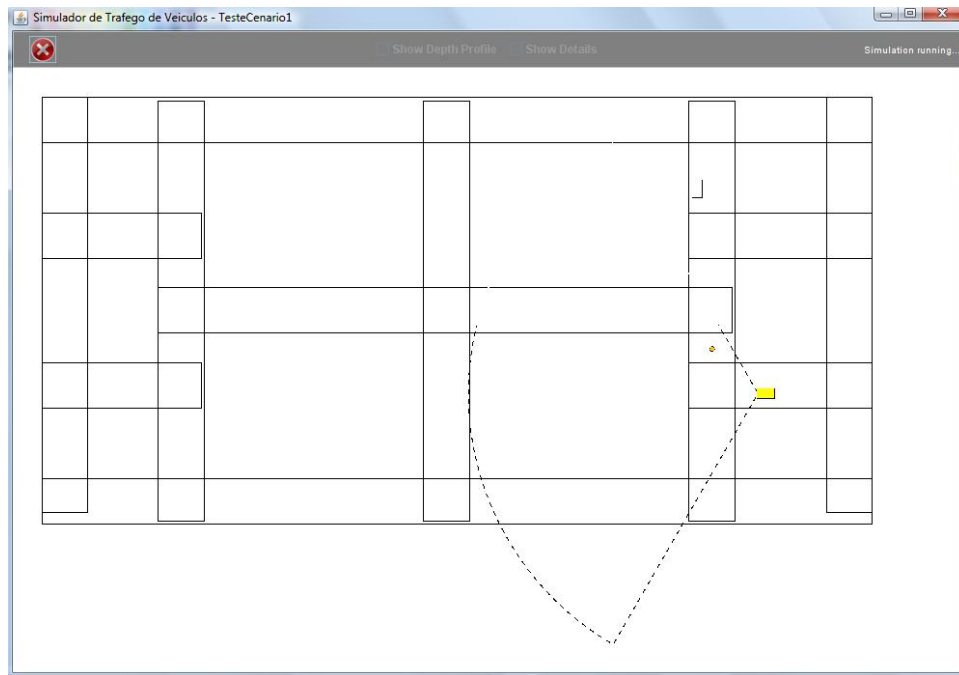


Figura 4.3: Vista global do cenário

diâmetro angular do objeto. Quando não há detecção de objeto algum, a informação é considerada *background*, que é delimitado pelas bordas direita e esquerda ou por algum objeto detectado. Por exemplo, a informação $background((leftBorder, ped1), 30, 4)$ significa que o intervalo sem obstáculos vai da borda esquerda até o objeto *ped1*, sua distância é trinta metros e sua abertura angular é de quatro graus.

4.5 Geração das Mensagens de Alerta

As previsões são feitas com base no DDPC (SOUTCHANSKI; SANTOS, 2008), descrito na seção 3.3, para saber a situação do objeto (*extending, shrinking, vanishing, appearing*). Com os dados da situação atual mais os de algumas situações do passado, é feita a previsão da posição do objeto em instantes futuros. Neste sistema é considerada apenas a situação *extending* por ser a única relevante para os alarmes de colisão, frenagem e ultrapassagem implementados em (FERRARI et al., 2008). A previsão é recalculada cada vez que o sensor do veículo detecta algum objeto, pois todos os envolvidos podem mudar seu percurso a qualquer momento.

Para classificar os riscos detectados em uma ou mais mensagens de alerta, são utilizadas retas virtuais. A figura 4.6 ilustra estas considerações geométricas. O retângulo vermelho representa a área do veículo e é utilizada nos cálculos para a geração da mensagem de colisão, que acontece

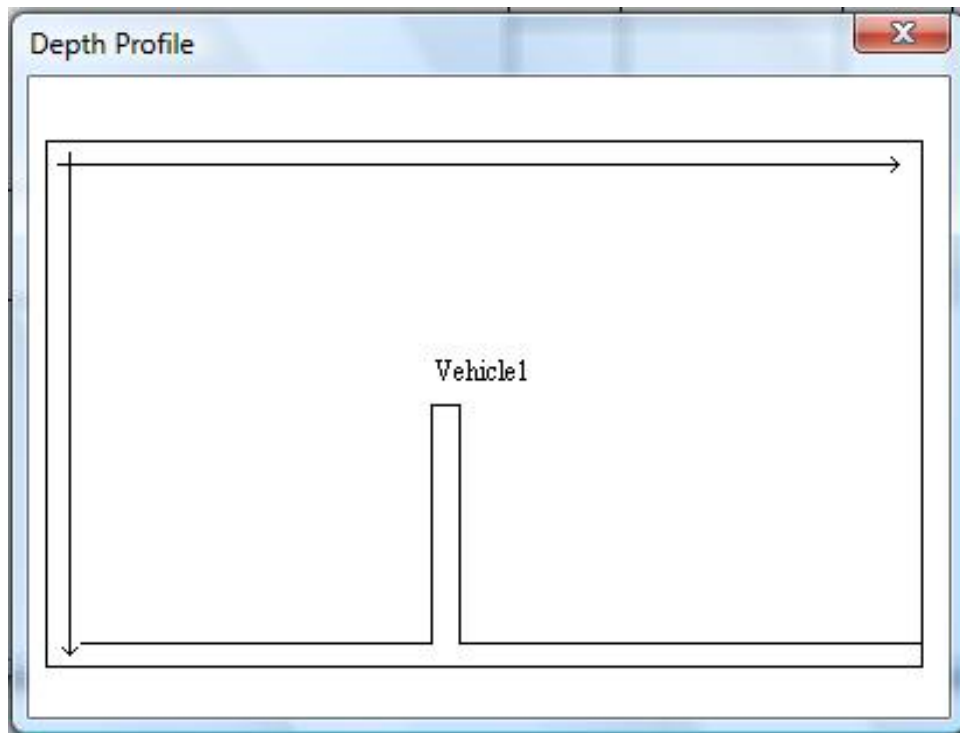


Figura 4.4: Gráfico de Perfil de Profundidade



Figura 4.5: Mensagens de alerta

quando a previsão dos estados futuros mostra que a posição do veículo nos instantes seguintes terá parte do objeto (representado pelo círculo) dentro de sua área. Já a mensagem de frenagem é gerada instantes antes, quando o objeto se aproxima da área vermelha mostrada na figura 4.6. Para a mensagem de ultrapassagem indevida, é feito o mesmo processo, mas as áreas analisadas são as das laterais do veículo, mostradas em amarelo.

No trabalho apresentado nesta dissertação foi necessário o acréscimo das situações descon- sideradas no simulador, uma vez que para a tomada de decisões, todas as informações disponíveis são indispensáveis na inferência dos estados futuros. O trabalho descrito em (FERRARI et al., 2008) envolve apenas uma implementação simplificada das regras e axiomas do DDPC, levando em consideração apenas a situação *extending*, que quando observada dispara as mensagens de alerta. Em contrapartida, no presente trabalho, foi gerada uma base de conhecimento mais sólida, com

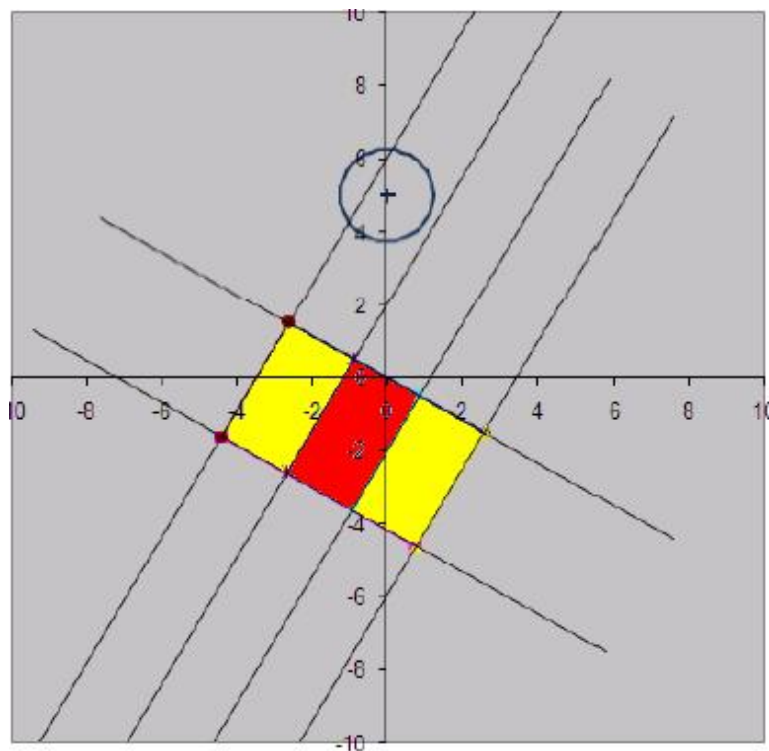


Figura 4.6: Áreas levadas em consideração na geração dos alertas
Fonte: (FERRARI et al., 2008)

axiomas definindo um conjunto de predicados que representem todas as situações em tráfego de veículos, conforme descrito no próximo capítulo.

5 FORMALIZAÇÃO

No presente capítulo será descrita a formalização desenvolvida nesta dissertação. Primeiro será explicado quais foram as ações selecionadas para fazer parte do sistema e como definiu-se cada uma delas. Em seguida estas ações são axiomatizadas, e por fim, descreve-se a implementação das regras em programação lógica.

5.1 Ações

Baseando-se em tráfego de veículos reais, foram investigadas quais ações um motorista realiza para chegar a um determinado destino em segurança. Além dos trabalhos publicados sobre o assunto (GERBER; NAGEL; SCHREIBER, 2002), (GALATA et al., 2002), (NARANJO et al., 2003), (NAGEL, 2004), (THRUN, 2006) e da observação pessoal pelo tráfego da cidade em dias de rotina ou durante viagens, também foram observados vídeos utilizados em pesquisas de tráfego de veículos autônomos.¹

Um dos pontos mais marcantes observados na interação entre os veículos, que é o foco deste estudo, foi que cada ação tomada baseia-se principalmente na distância que os veículos procuram manter uns dos outros nas vias. Com isto, foi possível definir qual seria a base das fórmulas que determinam os riscos e as ações para minimizá-los.

Partindo-se então da variável “distância”, nota-se que quanto mais próximos os veículos estão uns dos outros, maior o risco de acidentes. Isto afeta diretamente a variável “velocidade”, ou seja, quanto mais os veículos se aproximam, mais eles tendem a diminuir a velocidade de deslocamento. Porém, atualmente, percebe-se que o motorista reduz a velocidade do veículo somente quando esta é a última alternativa. É neste ponto que entram as variáveis referentes à mudança de faixa, necessárias às ultrapassagens ou desvio de obstáculos. Quando existe um obstáculo à frente e não é possível mudar de faixa à esquerda e nem à direita, não resta outra alternativa a não ser parar e aguardar uma possibilidade de desvio. Para isso foi considerada uma distância mínima entre veículos que, quando atingida, habilita uma variável que permite que o veículo reinicie movimento

¹Os vídeos citados são filmagens de eventos como o Darpa Challenge e Tartan Racing. Disponível em <http://www.youtube.com/>. Acesso em: 23 Abr. 2011.

após uma solicitação de parada do sistema.

Ao serem definidas as ações possíveis conforme descrito abaixo, foram desenvolvidos axiomas para formalizar tais ações que serão explicadas na próxima seção. Estes são axiomas que definem as restrições para o tráfego seguro de veículos. Isto é, não foram projetados para o controle automático de um veículo autônomo, mas sim para agir como um sistema coadjuvante.

5.2 Axiomas

Os axiomas que representam logicamente as ações do Sistema de Segurança de Tráfego de Veículos estão descritos a seguir, onde v é uma variável para o veículo inteligente, t é uma variável para o instante de tempo e o é a variável para veículos comuns. A variável δ_n é um valor genérico em *um* que representa as distâncias de segurança para cada ação, podendo seus valores serem alterados caso o sistema seja implementado em um veículo real ou qualquer outro sistema de auto-locomoção, respeitando-se sempre a proporcionalidade existente no código. A variável θ representa os graus ocupados no cone de visão do veículo inteligente. Para todos os axiomas foi considerado apenas um veículo autônomo interagindo com veículos dirigidos por pessoas.

A ação *Parar* é inferida pelo sistema quando a distância entre objeto e observador mostrada no pico gerado pelo Cálculo de Perfil de Profundidade atinge o valor mínimo δ_1 determinado para segurança. Como foi considerado apenas um veículo inteligente, o risco de dois veículos identificarem a distância mínima de segurança, pararem e não se movimentarem mais é descartada pois antes do veículo inteligente parar, entram em ação as regras de desvio e considera-se que no veículo não inteligente haja um condutor capaz de reiniciar movimento e desviar caso o veículo inteligente pare. Esta situação é representada pelo axioma [A1].

$$[A1] \text{ Parar}(v, o, t) \leftarrow \text{dist}(v, o, t) \leq \delta_1.$$

Após uma solicitação de parada, a ação *Reiniciarmov* (axioma [A2]) permite que o veículo inteligente volte a se movimentar e continuar seu percurso. Esta ação é inferida assim que a distância entre o veículo inteligente e o objeto a sua frente saia do nível inseguro, ou seja, a distância entre eles deve ser maior do que δ_1 . Esta ação é sugerida principalmente em cruzamentos, quando o veículo inteligente para e aguarda os veículos comuns passarem para só então seguir adiante seu percurso. Também pode ser aplicada a veículos em fila aguardando o sinal verde dos semáforos. Caso um veículo comum à frente do veículo inteligente pare e não exista a pos-

sibilidade de desvio, o veículo inteligente vai permanecer parado até o obstáculo à sua frente se movimentar. A situação de semáforos não foi testada por não haver esta opção no simulador.

$$[A2] \text{ Reiniciarmov}(v, o, t_2) \leftarrow \text{dist}(v, o, t_2) > \delta_1 \wedge \text{Parar}(v, o, t_1), \wedge \neg \text{Acelerar}(v, o, t_2) \wedge \\ \neg \text{Mantervelcte}(v, o, t_2) \wedge \neg \text{Desacelerar}(v, o, t_2) \wedge (t_2 > t_1).$$

Se a distância entre o veículo inteligente e o objeto estiver entre δ_1 e δ_2 e o veículo estiver em movimento, há duas opções de ações a serem determinadas: *Mudarfd* (axioma [A4] mudar para a faixa da direita) ou *Mudarfe* (axioma [A3] mudar para a faixa da esquerda). A decisão é tomada levando-se em consideração o espaço angular ocupado pelo objeto no cone de visão do observador, dado este também gerado pelo pico do Cálculo de Perfil de Profundidade. Em uma via, o sistema compara o espaço livre do lado esquerdo θ_e e do lado direito θ_d do objeto, mudando de faixa para o lado com maior espaço, analisando se não há nenhum outro objeto na faixa de destino e se o espaço apresentado é suficiente para sua passagem. A distância considerada para a mudança de faixa com o objetivo de desvio de um obstáculo, que pode ser um veículo parado ou movimentando-se com velocidade menor do que a do veículo inteligente, contém a distância considerada para reiniciar movimento, sendo assim, é premissa para um desvio não haver uma solicitação de parada do sistema no tempo anterior ao considerado para o desvio. Se for identificada uma situação de parada no tempo anterior ao consultado, necessariamente o sistema deve considerar o reinício de movimento. Neste trabalho não utilizamos sistemas para detecção de faixas de rodagem pois já existem trabalhos desenvolvidos especificamente nesta área (HU; TAO; HU, 2004), (NAGEL, 2004), (GERBER; NAGEL; SCHREIBER, 2002), (HUMMEL, 2009), (MCCALL; TRIVEDI, 2006), não sendo este nosso foco.

$$[A3] \text{ Mudarfe}(v, o, t_2) \leftarrow \delta_1 < \text{dist}(v, o, t_2) \leq \delta_2 \wedge \text{ang}(v, o, t_2, \theta_d, \theta_e) \wedge (\theta_e > \theta_d) \wedge \\ (\theta_e > \delta_5) \wedge \neg \text{Parar}(v, o, t_1) \wedge (t_2 > t_1);$$

$$[A4] \text{ Mudarfd}(v, o, t_2) \leftarrow \delta_1 < \text{dist}(v, o, t_2) \leq \delta_2 \wedge \text{ang}(v, o, t_2, \theta_d, \theta_e) \wedge (\theta_e < \theta_d) \wedge \\ (\theta_d > \delta_5) \wedge \neg \text{Parar}(v, o, t_1) \wedge (t_2 > t_1).$$

As ações referentes ao controle de velocidade estão diretamente relacionadas à distância objeto-observador apresentada pelos picos. Quanto maior a distância entre eles, mais seguro o sistema admite estar o veículo inteligente para acelerar (axioma [A7]). Conforme esta distância vai diminuindo, o sistema considera prudente manter a velocidade constante (axioma [A6]) e se o pico continuar apresentando o mesmo comportamento, desacelerar (axioma [A5]).

[A5] $Desacelerar(v, o, t_2) \leftarrow \delta_2 < dist(v, o, t_2) \leq \delta_3 \wedge \neg Parar(v, t_1) \wedge (t_2 > t_1);$

[A6] $Mantervelcte(v, o, t_2) \leftarrow \delta_3 < dist(v, o, t_2) \leq \delta_4 \wedge \neg Parar(v, t_1) \wedge (t_2 > t_1);$

[A7] $Acelerar(v, o, t_2) \leftarrow dist(v, o, t_2) > \delta_4 \wedge \neg Parar(v, t_1) \wedge (t_2 > t_1).$

Quando o veículo inteligente percorre a via e nenhum objeto entra em seu cone de visão, nenhum pico é apresentado e a resposta do sistema será que o agente está habilitado a acelerar. Isso não quer dizer que o veículo vá acelerar continuamente até que um obstáculo entre em seu campo de visão, mas significa que, caso necessário, o veículo pode acelerar pois não colidirá com nenhum objeto naquele instante. Como este é um sistema coadjuvante, deve ser utilizado em conjunto com outros sistemas, por exemplo, que identifiquem placas de sinalização que contenham a informação de velocidade máxima permitida e produzam no agente a ação de controle de velocidade. Desta maneira, é possível criar uma regra onde o veículo seja habilitado a acelerar até atingir a velocidade permitida na via, entrando depois na condição de manter velocidade constante.

A partir do momento em que um objeto é identificado pelo sistema, a distância entre eles passa a ser monitorada através dos picos apresentados quadro a quadro. Quando esta distância entra nos valores considerados de risco, são habilitadas as regras para manter velocidade constante, diminuir velocidade, mudar para faixa da esquerda ou direita e então parar. A regra para reiniciar o movimento só é habilitada quando a distância entre agente e objeto é suficiente para uma movimentação sem risco.

A implementação destes axiomas foi feita em PROLOG, conforme explicado na seção seguinte.

5.3 Implementação

Para a implementação das regras de ações contidas no Sistema de Segurança em Tráfego de Veículos foi escolhida uma linguagem de programação lógica denominada PROLOG.

PROLOG é uma linguagem de programação declarativa desenvolvida nas décadas de 60 e 70 (PEREIRA; SHIEBER, 2005) que pode ser implementada em um interpretador de código livre muito popular em Inteligência Artificial denominado SWI-PROLOG (WIELEMAKER, 2003).

Os programas estruturados em PROLOG expressam se existe ou não uma relação entre determinadas entidades em que condição isso acontece (PEREIRA; SHIEBER, 2005). São normalmente escritos em subconjuntos de lógica de primeira ordem (cláusulas de Horn), onde as enti-

dades envolvidas são consideradas constantes e as relações entre estas entidades são denominadas predicados.

Um programa em PROLOG consiste em um conjunto de axiomas formados por entidades e suas relações escritos em forma de cláusulas para expressar fatos que formam uma base de conhecimento a ser consultada através das *queries*.

A base de conhecimento pode conter cláusulas simples ou regras, onde as regras substituem uma exaustiva lista de cláusulas simples para determinar se alguma relação descrita é verdadeira (PEREIRA; SHIEBER, 2005).

Como o programa desenvolvido neste trabalho contém uma série de fatos obtidos de um domínio e regras de segurança baseadas em lógica, esta linguagem permite executar buscas onde apresentam-se os fatos através de uma base de conhecimento e espera-se receber como resposta uma ação cuja regra seja adequada à situação mostrada naquele instante.

A sequência de fatos é apresentada ao PROLOG através uma base de conhecimento. Esta base é gerada com os dados identificados no Cálculo de Perfil de Profundidade Dinâmico (seção 3.3), utilizando-se o Simulador de Tráfego de Veículos (capítulo 4). Para gerar os dados, primeiramente define-se uma cena que deve conter o veículo inteligente e no mínimo um veículo comum. Especifica-se também os respectivos percursos que cada veículo deve percorrer, tanto para o agente quanto para os objetos. Ao simular os movimentos executados pelos veículos, o simulador gera os picos que representam os objetos presentes no cone de visão do agente e salva um arquivo contendo as características como tamanho, profundidade e distância do agente, separadas quadro a quadro. Estes dados são adaptados para assumir a seguinte forma:

descricao(objeto,tempo,distancia,angulodireito,anguloesquerdo).

A base de conhecimento é formada por uma série de fatos descritos conforme mostrado acima, onde a variável objeto representa um veículo identificado no cone de visão do agente, nomeado como “veículo1”, “veículo2”, e assim sucessivamente até o último objeto passar pela visão do observador. Se um mesmo objeto passar pelo agente em instantes diferentes e por percursos diferentes, o sistema o reconhece e o identifica na base de conhecimento com o mesmo índice com que foi descrito inicialmente.

A variável tempo refere-se ao instante ou quadro em que a informação foi gerada, sendo apresentada ao sistema em ordem cronológica.

A variável distância determina à qual distância encontra-se o objeto do agente. É através desta variável associada às informações relativas ao tempo que o sistema identifica o comporta-

mento dos veículos e do agente, inferindo se eles estão se aproximando ou distanciando, para então sugerir a ação recomendada para o instante.

As variáveis ângulo direito e ângulo esquerdo referem-se a quantos graus do cone de visão do veículo inteligente estão livres ao lado direito do objeto identificado e ao lado esquerdo do mesmo. Com estes dados analisa-se uma possível mudança de faixa quando necessário. Os graus ocupados no cone visão são considerados a largura das entidades envolvidas, assim sendo, considerou-se que a via ocupa trinta graus do cone de visão. Cada veículo foi padronizado para ocupar aproximadamente dez graus, inclusive o agente. Para assumir uma mudança de faixa o sistema tem a largura total da via, retirando o valor do veículo observado e analisando se os graus restantes do lado direito e esquerdo da via são suficientes para sua passagem segura, ou seja, o espaço deve ser maior do que doze graus.

Na seção seguinte serão apresentadas as regras implementadas em PROLOG.

5.4 Regras em PROLOG

Adiante tem-se os axiomas descritos na seção 5.2 transformados em regras PROLOG.

Como a principal variável que comanda as regras é a distância, o programa foi organizado com o valor desta variável em ordem crescente, uma vez que o PROLOG executa a leitura de cima para baixo.

Para a situação de maior risco, a distância é a menor, determinada em $2um$. Com este valor ainda é possível observar-se no simulador uma pequena distância entre objeto e agente e o pico gerado ainda traz informação de distância antes de perder a informação, uma vez que o sistema não opera ações no veículo para que este pare ou desvie. A seguinte regra representa o axioma [A1] apresentado na seção 5.2.

$$\begin{aligned} situacao(objeto, tempo, parar) : - \\ \quad descricao(objeto, tempo, profund, -, -), \\ \quad profund = < 2. \end{aligned}$$

Para que o veículo esteja apto a reiniciar movimento após uma parada (ação *reiniciarmov*), é necessário que a distância entre agente e objeto seja maior do que $2 um$. Para habilitar esta regra,

também é necessário que no instante anterior o pico mostrado tenha disparado uma resposta de parada no sistema. A regra abaixo representa o axioma [A2].

$$\begin{aligned}
 & \textit{situacao}(\textit{objeto}, \textit{tempo}, \textit{reiniarmov}) : - \\
 & \quad \textit{descricao}(\textit{objeto}, \textit{tempo}, \textit{profund}, -, -), \\
 & \quad \textit{Tempoanterior} \textit{ is } \textit{tempo} - 1, \\
 & \quad \textit{profund} > 2, \\
 & \quad \textit{situacao}(\textit{objeto}, \textit{Tempoanterior}, \textit{parar}), \\
 & \quad \textit{not}(\textit{situacao}(\textit{objeto}, \textit{tempo}, \textit{acelerar})), \\
 & \quad \textit{not}(\textit{situacao}(\textit{objeto}, \textit{tempo}, \textit{desacelerar})), \\
 & \quad \textit{not}(\textit{situacao}(\textit{objeto}, \textit{tempo}, \textit{matermovcte})).
 \end{aligned}$$

As distâncias definidas para desvio de objetos ficaram entre 2 e 3 *um*. Antes de dar uma resposta de parada, o veículo procura desviar do objeto que se movimenta mais lentamente ou está totalmente parado. O sistema analisa qual a direção que oferece maior espaço restante na via e se este espaço é maior do que a dimensão do agente, fixada em 10 graus. Por segurança, definiu-se 12 graus para habilitar a passagem do agente. Abaixo temos duas regras, a primeira define o axioma [A3], para a ação *mudarparafaixadaesquerda*. A segunda define o axioma [A4], para a ação *mudarparafaixadadireita*.

$$\begin{aligned}
 & \textit{situacao}(\textit{objeto}, \textit{tempo}, \textit{mudarfe}) : - \\
 & \quad \textit{descricao}(\textit{objeto}, \textit{tempo}, \textit{profund}, \textit{Angdir}, \textit{Angesq}), \\
 & \quad \textit{Tempoanterior} \textit{ is } \textit{tempo} - 1, \\
 & \quad \textit{profund} = < 3, \\
 & \quad \textit{profund} > 2, \\
 & \quad \textit{Angesq} \geq \textit{Angdir}, \\
 & \quad \textit{Angesq} > 12, \\
 & \quad \textit{not}(\textit{situacao}(\textit{objeto}, \textit{Tempoanterior}, \textit{parar})).
 \end{aligned}$$

situacao(objeto,tempo,mudarfd) : –

descricao(objeto,tempo,profund,Angdir,Angesq),
Tempoanterior is tempo – 1,
profund = < 3,
profund > 2,
Angesq < Angdir,
Angdir > 12,
not(situacao(objeto,Tempoanterior,parar)).

Antes de qualquer ação de desvio ou parada, assim que um objeto é identificado e gerado um pico, inicia-se um monitoramento de velocidade. Quando a distância entre agente e objeto está entre 3 e 5 *um*, o sistema sugere que o veículo desacelere (ação *desacelerar*). Caso o pico continue crescendo e mostrando que a distância entre eles continua diminuindo, o sistema entra nas regras de mudança de faixa com o objetivo de desvio. A regra abaixo representa o axioma [A5].

situacao(objeto,tempo,desacelerar) : –

descricao(objeto,tempo,profund,-,-),
Tempoanterior is tempo – 1,
profund = < 5,
profund > 3,
not(situacao(objeto,Tempoanterior,parar)).

Com a distância entre 5 e 7 *um* o veículo inteligente será habilitado a manter sua velocidade constante (ação *mantervelcte*). Se com este comportamento no instante seguinte o pico aumentar, mostrando uma diminuição na distância do objeto, o sistema vai sugerir a diminuição da velocidade, se o pico diminuir, mostrando o aumento da distância, o veículo estará habilitado a acelerar. A regra abaixo representa o axioma [A6].

situacao(objeto,tempo,mantervelcte) : –

descricao(objeto,tempo,profund,-,-),
Tempoanterior is tempo – 1,
profund > 5,
profund = < 7,
not(situacao(objeto,Tempoanterior,parar)).

Com a distância entre objeto e agente maior do que 7 *um*, o veículo está habilitado a *acelerar* sem riscos de colisão, conforme representado pela regra abaixo (que implementa o axioma [A7]).

$$\begin{aligned}
 & \textit{situacao}(\textit{objeto}, \textit{tempo}, \textit{acelerar}) : - \\
 & \quad \textit{descricao}(\textit{objeto}, \textit{tempo}, \textit{profund}, -, -), \\
 & \quad \textit{Tempoanterior} \textit{ is } \textit{tempo} - 1, \\
 & \quad \textit{profund} > 7, \\
 & \quad \textit{not}(\textit{situacao}(\textit{objeto}, \textit{Tempoanterior}, \textit{parar})).
 \end{aligned}$$

Todas estas regras foram testadas e os resultados destes testes serão apresentados no capítulo seguinte.

6 TESTES

Os testes foram realizados através do Simulador de Tráfego de Veículos (capítulo 4), onde diversas situações de risco foram apresentadas ao sistema. Em um mesmo cenário estavam presentes o veículo inteligente e veículos comuns que serviram de obstáculos ao agente. O percurso de todos os envolvidos foi pré-determinado pelo usuário com o objetivo de inferir o percurso do veículo inteligente e desta maneira testar as ações sugeridas pelo sistema.

Não foram consideradas se as vias são de mão única ou dupla, nem uma quantidade fixa de faixas, uma vez que o sistema analisa a largura total da via e sugere desvios para a direita ou para a esquerda. Considerou-se também que todos os carros possuem a mesma dimensão física, ou seja, ocupam a mesma quantidade de graus da visão do observador. Também não foi considerada visão da parte traseira do veículo inteligente, visão esta que seria equivalente aos espelhos retrovisores. Na realização dos testes o veículo inteligente interagiu apenas com outros veículos, sendo assim, pedestres, animais ou qualquer outro obstáculo foram desconsiderados. Os veículos comuns realizaram percursos coerentes, de um ponto inicial a um final, não sendo aleatória sua movimentação.

Após o posicionamento dos veículos e determinação dos percursos, habilitou-se o Simulador de Tráfego de Veículos para gerar os movimentos e os perfis de profundidade dinâmicos como no exemplo mostrado na Figura 6.2. Os perfis foram gerados quadro a quadro, e transformados em uma fórmula lógica, de onde extraiu-se a base de conhecimento para o Sistema de Segurança em Tráfego de Veículos. Adiante, nas formulas F1 a F13 é mostrado como exemplo uma parte da base de conhecimento gerada pela simulação do cenário mostrado na Figura 6.1. Neste cenário as linhas representam o percurso que cada integrante da cena deve percorrer. A simulação gerou mais fórmulas do que as descritas no exemplo a seguir, formando uma base de conhecimento bem maior, porém foram omitidas por brevidade.

Como é possível identificar tanto através dos picos quanto através dos valores na base de dados, primeiro o agente identifica o veículo 1, bem próximo de si como mostrado na Figura 6.1, demonstrado através de um pico alto na Figura 6.2 e identificado pelo sistema como “veiculo1” na base de conhecimento. Após alguns instantes, o agente identifica o veículo 2, que no instante inicial encontrava-se mais distante do agente e fora do cone de visão, conforme cenário da Figura 6.1, representado por um pico menor na Figura 6.2, identificado pelo sistema como “veiculo2”,

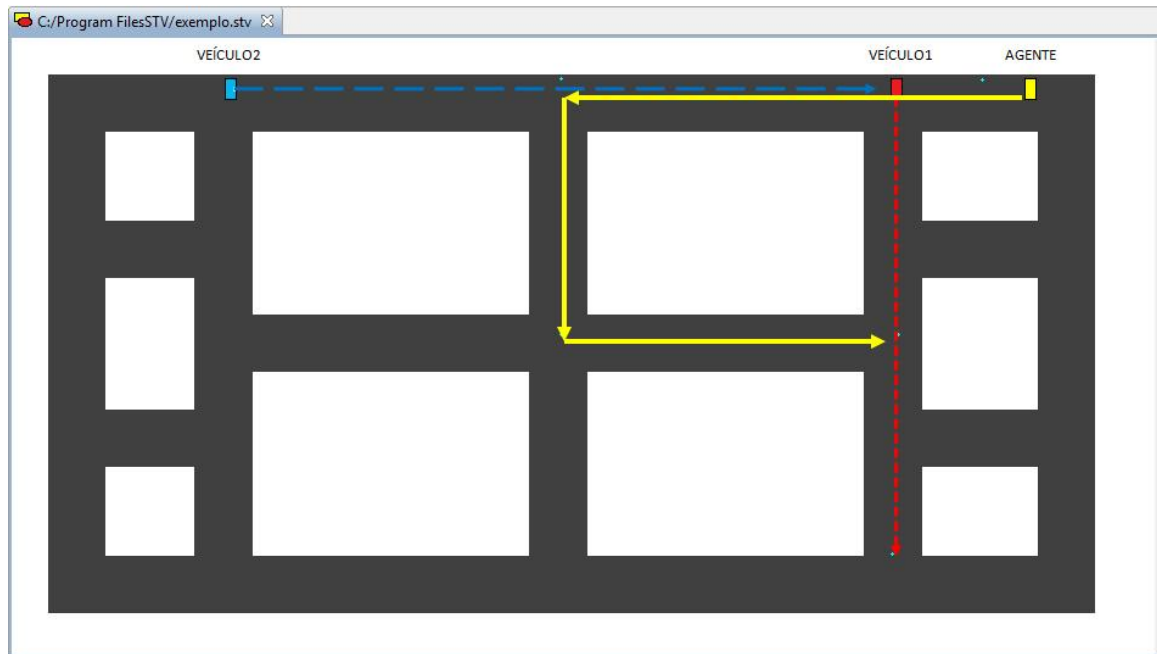


Figura 6.1: Cenário montado com agente e dois veículos comuns para exemplo.

tendo seu pico aumentando de tamanho conforme o veículo aproxima-se do agente.

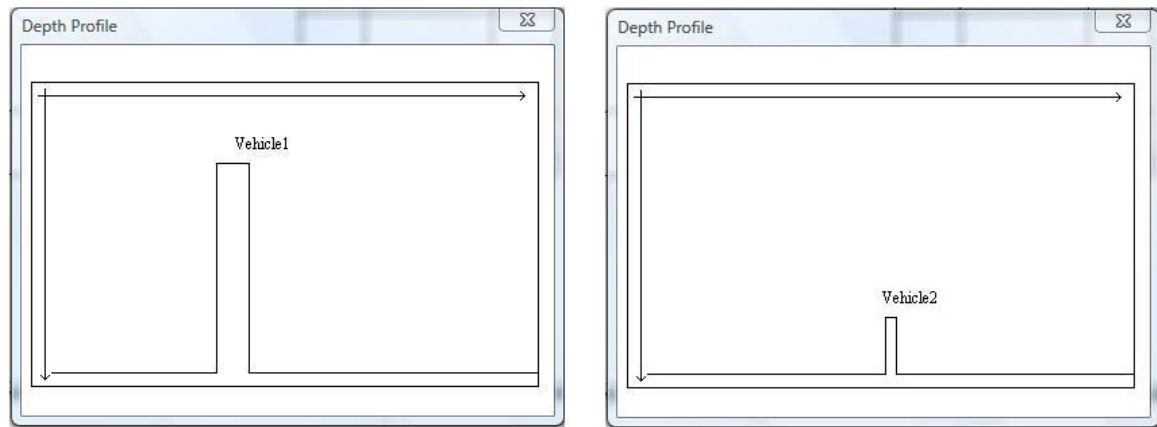


Figura 6.2: Imagem dos picos que representam os veículos 1 e 2 em determinado instante da simulação.

Um subconjunto da base de conhecimento gerada para a cena da Figura 6.1 está descrito nas formulas F1 a F13 a seguir, lembrando que conforme descrito no capítulo 5, deve ter o formato

descricao(objeto,tempo,distancia,angulodireito,anguloesquerdo).

[F1] *descricao(veiculo1,0,19.2,2.2,21.1).*

[F2] *descricao(veiculo1,1,17.3,2.2,20.1).*

[F3] *descricao(veiculo1,2,15.8,2.2,18.6).*

[F4] *descricao(veiculo1,3,14.6,2.2,17.9).*

[F5] *descricao(veiculo1,4,13.8,2.2,18.0).*

[F6] *descricao(veiculo1,5,12.6,2.2,17.5).*

[F7] *descricao(0,8,100.0,2.0,28.0).*

[F8] *descricao(veiculo2,18,56.3,2.2,25.4).*

[F9] *descricao(veiculo2,19,52.1,2.2,25.0).*

[F10] *descricao(veiculo2,20,48.0,2.2,24.5).*

[F11] *descricao(veiculo2,21,44.6,2.2,24.0).*

[F12] *descricao(veiculo2,22,40.9,2.2,23.5).*

[F13] *descricao(veiculo2,23,36.5,2.2,23.0).*

A base de conhecimento descreve que o primeiro veículo a entrar no cone de visão do agente é identificado como "veiculo1" e encontra-se 19.2um distante do agente, deixando livre na via 21.1 graus ao lado esquerdo do veículo identificado e 2.2 graus do lado direito, conforme fórmula [F1]. No instante seguinte o sistema identifica o veículo a uma distância de aproximadamente 17um, deixando livre na via 20.1 graus ao lado esquerdo do veículo identificado e 2.2 graus do lado direito (fórmula [F2]). A cada instante a distância mostrada vai diminuindo, representando a aproximação entre agente e veículo comum. Quanto mais se aproximam, menor o ângulo restante na visão do observador e maior o ângulo ocupado pelo veículo que se aproxima. Alguns instantes depois o sistema não identifica veículo algum em seu caminho, assumindo a fórmula padrão [F7]. Em seguida, é identificado o veículo2 na fórmula [F8], a uma distância de 56.3um, explicação para o pequeno pico mostrado na Figura 6.2. Seguindo a mesma linha de raciocínio, conforme aproximam-se os veículos, menor o ângulo restante no cone de visão do agente, e mais restritas as possibilidades de desvio.

As bases de conhecimento geradas pela simulação dos diversos cenários testados foram inseridas no programa desenvolvido em PROLOG (seção 5.3), onde consta o código com as fórmulas que representam os axiomas explicados na seção 5.2. Foram gerados dez cenários com diferentes disposições de veículos e percursos, e conseqüentemente, diferentes bases de conhecimento. Cada base de conhecimento contém pelo menos seis situações de risco para o agente.

A tabela abaixo demonstra as situações possíveis, a quantidade de vezes em que estas

situações apareceram nos diferentes cenários testados e a porcentagem de acertos.

Situação	Cenários	% de acertos
Parar	10	100
Reiniciar movimento	8	100
Mudar para faixa da direita	6	100
Mudar para faixa da esquerda	6	100
Desacelerar	10	100
Manter movimento constante	10	100
Acelerar	10	100

Tabela 6.1: Tabela do total de situações simuladas analisadas no experimento

As consultas foram realizadas tanto em ordem cronológica dos fatos simulados quanto aleatoriamente através dos instantes em que os perfis de profundidade foram registrados. Após a simulação é gerada uma base de dados estática, portanto, as consultas realizadas através do PRO-LOG não precisam ser feitas necessariamente em ordem cronológica, podendo consultar-se qual a resposta do sistema para qualquer instante simulado. Foram geradas situações em vários graus de risco para o veículo inteligente e em seguida o sistema foi consultado, identificando cada uma delas, até mesmo as mais repentinas, como um veículo comum atravessando um cruzamento em alta velocidade. No simulador de tráfego de veículos é possível alterar a velocidade de movimento tanto do agente quanto dos veículos comuns, permitindo que as situações de mudança de faixa para desvio de veículos mais lentos fosse testada além dos desvios de veículos parados.

O sistema alerta para uma necessidade de parada quando na simulação a cena observada é similar à da Figura 6.3. A distância entre os veículos é a menor distância que possibilita uma parada segura.

Após uma solicitação de parada, para que o agente possa reiniciar movimento, a distância necessária é equivalente à mostrada na Figura 6.4.

Quando o agente encontra-se a uma distância como a mostrada na Figura 6.5, o sistema sugere que ele mova-se para a faixa da direita.

Quando o agente encontra-se a uma distância como a mostrada na Figura 6.6, o sistema sugere que ele mova-se para a faixa da esquerda.

A ação desacelerar é solicitada quando a situação do instante é igual à mostrada na Figura 6.7.

A Figura 6.8 mostra a distância segura para manter a velocidade constante.

A Figura 6.9 demonstra que entre o veículo inteligente e o veículo comum existe espaço

suficiente para acelerar.

A figura 6.10 representa um mesmo pico em diferentes instantes durante o distanciamento entre o veículo inteligente e um veículo comum. O pico é dinâmico e altera suas dimensões a cada instante em que os veículos em cena se movimentam, desde que estejam no campo de visão do agente. Quanto mais próximo o objeto está do observador, maior o pico naquele instante, conforme o objeto vai se distanciando, o pico vai diminuindo de tamanho e largura.

Para cada instante consultado no Sistema de Segurança em Tráfego de Veículos através do PROLOG a ação solicitada ao veículo inteligente foi condizente com a situação simulada. Em todos os casos apresentados os axiomas mostraram-se eficazes em evitar possíveis acidentes.

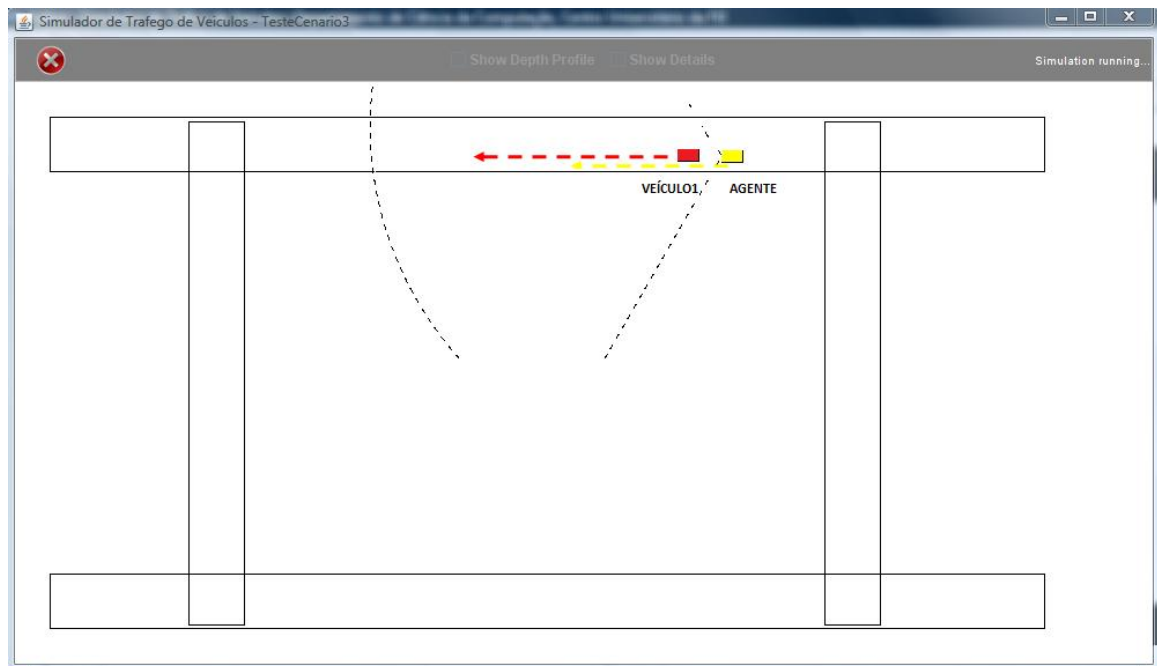


Figura 6.3: Cena gerada durante a simulação de uma situação onde o sistema alerta a necessidade de parada do veículo inteligente.

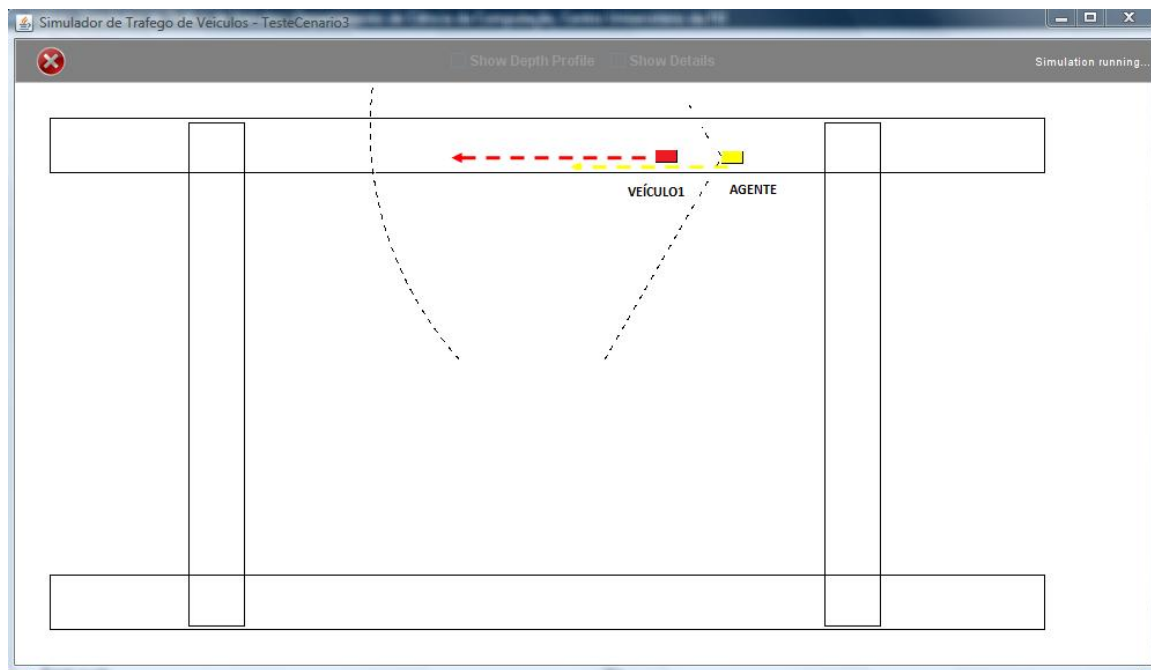


Figura 6.4: Distância de segurança necessária para o sistema habilitar o agente a reiniciar movimento após uma parada.

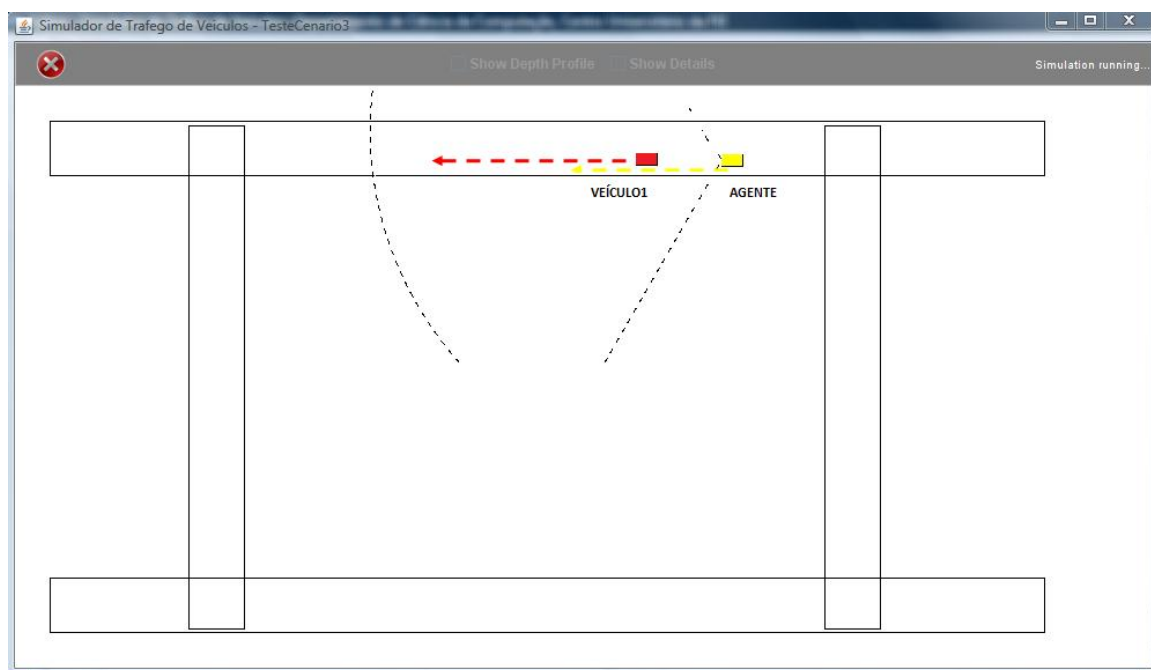


Figura 6.5: Distância segura para ultrapassagem ou desvio de obstáculo pela faixa da direita.

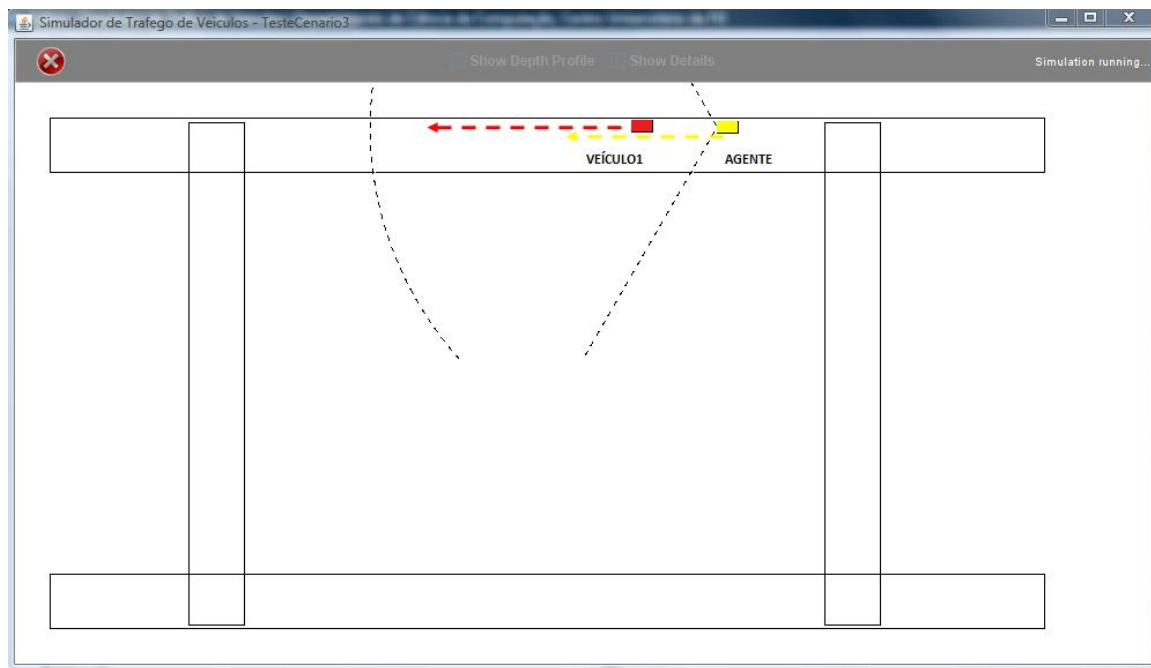


Figura 6.6: Distância segura para ultrapassagem ou desvio de obstáculo pela faixa da esquerda.

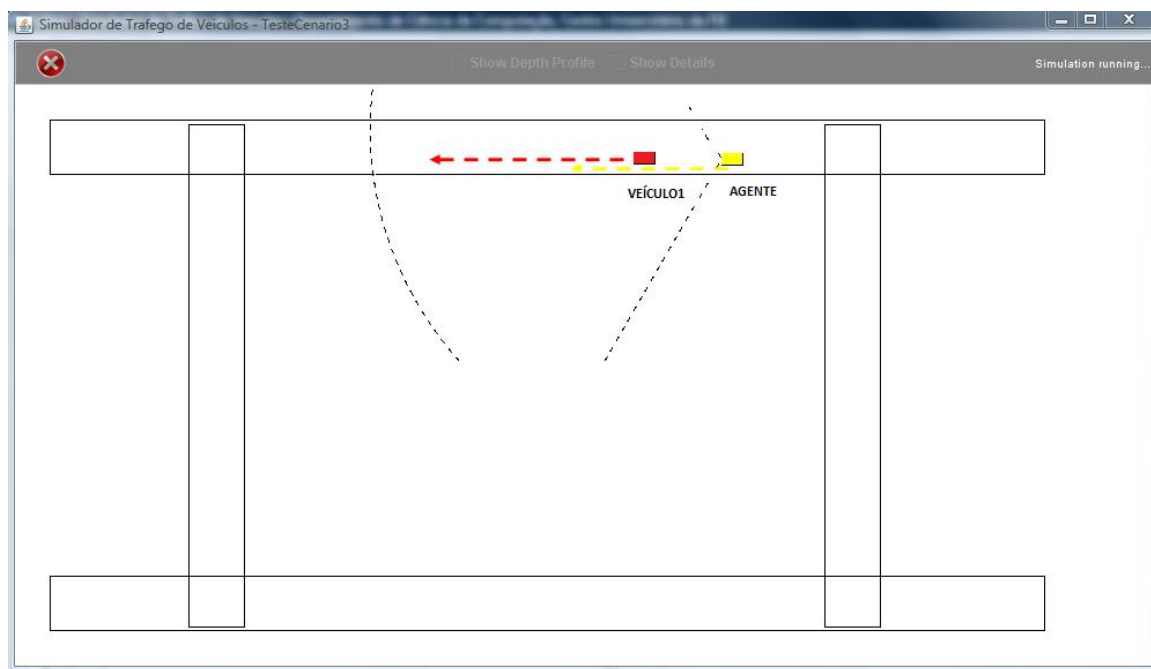


Figura 6.7: A cena mostra que os veículos estão próximos sendo conveniente que o agente comece a desacelerar.

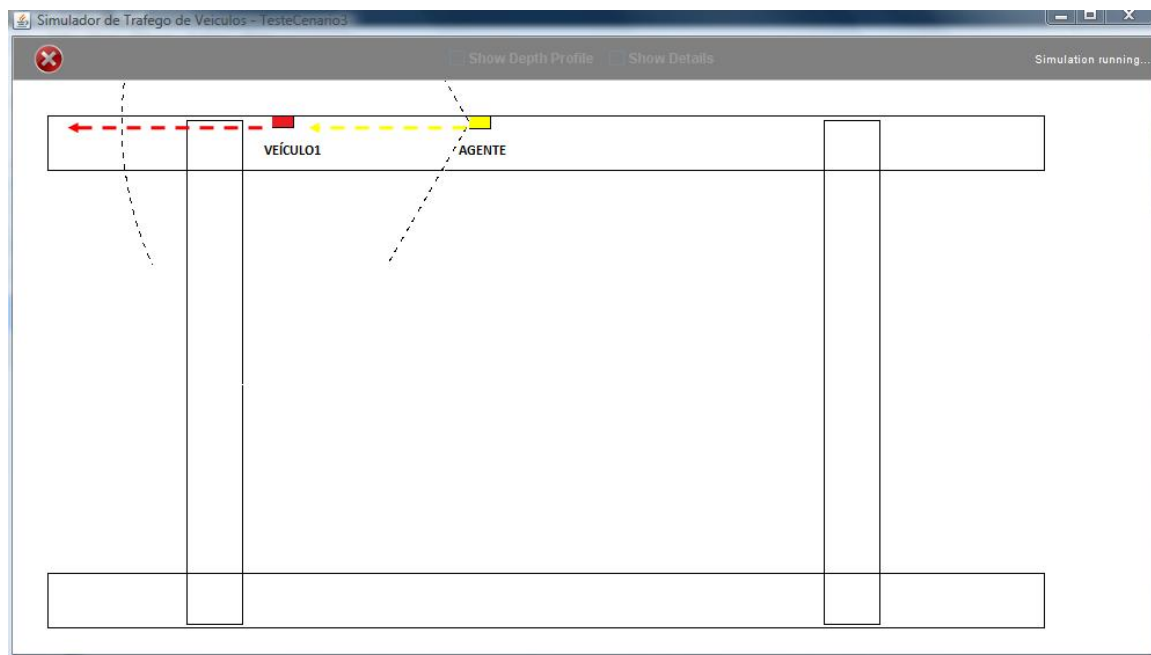


Figura 6.8: A distância entre os veículos ainda não demonstra grandes riscos mas considera-se prudente a partir deste ponto manter a velocidade constante.

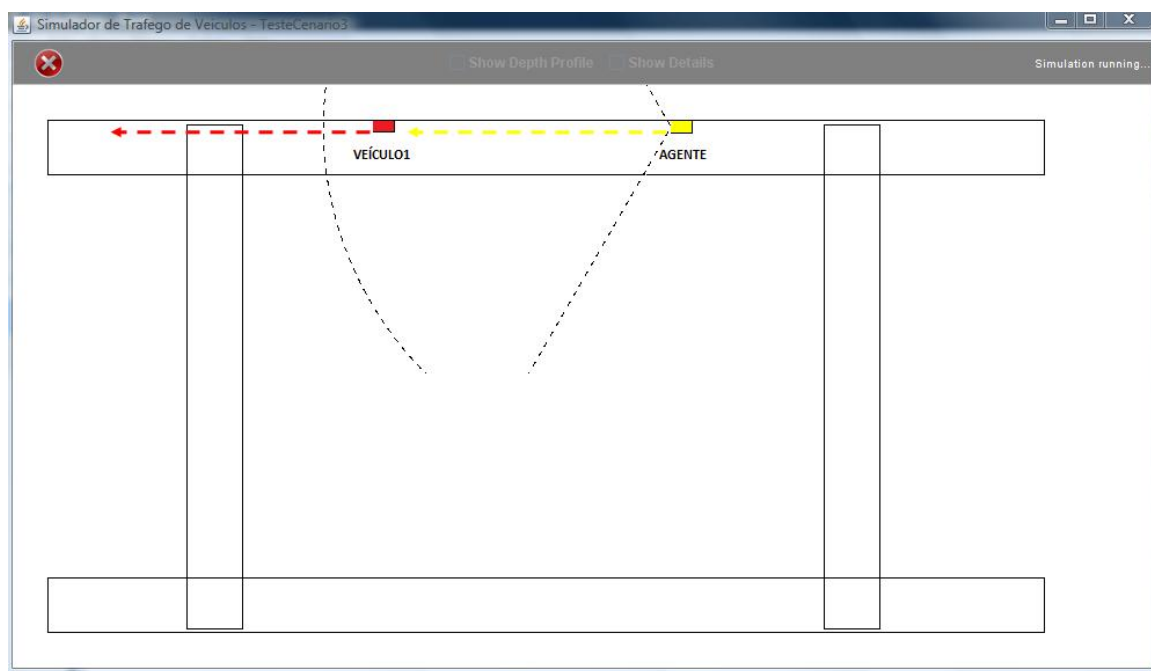


Figura 6.9: A distância entre os veículos é grande o suficiente para permitir que o veículo inteligente acelere.

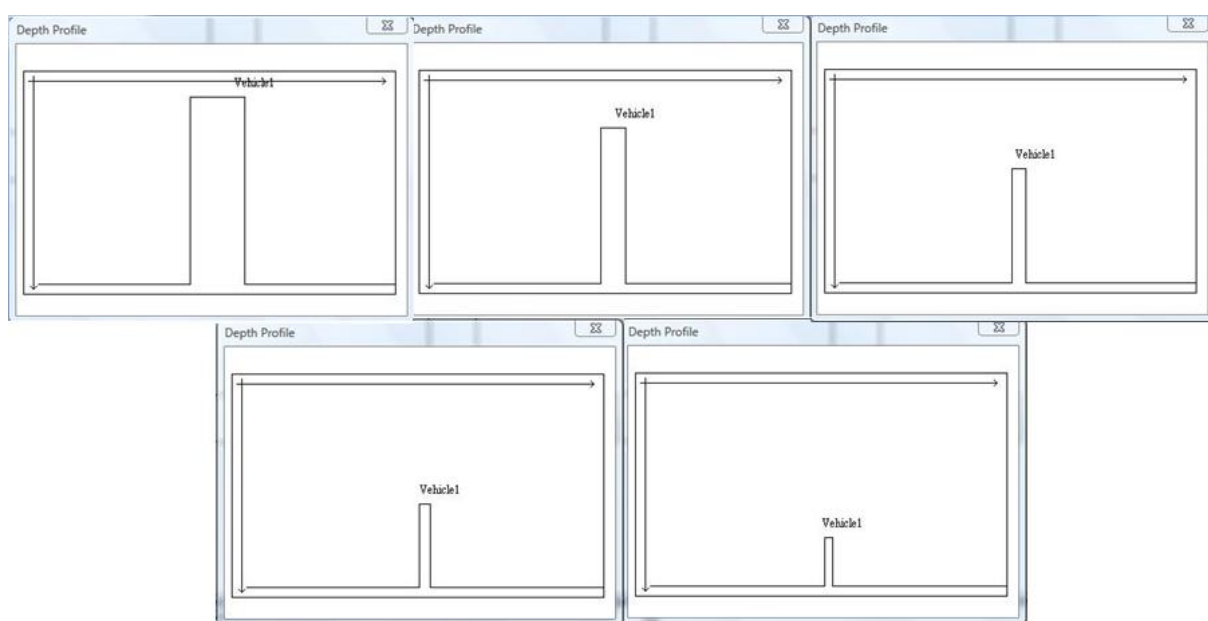


Figura 6.10: Picos representando o distanciamento entre o veículo inteligente e um veículo comum.

7 CONCLUSÃO

O Sistema de Segurança em Tráfego de Veículos proposto nesta dissertação mostrou-se eficiente nos testes realizados. Este sistema destaca-se por sua simplicidade e facilidade de implementação. Os dados de entrada podem ser adquiridos através de uma câmera estéreo, gerando informações suficientes para que o sistema possa inferir fatos não triviais sobre situações de tráfego. Além disso, a implementação não exige nenhum tipo de software ou hardware sofisticado.

O objetivo de utilizar os dados do pico gerado pelo Cálculo de Perfil de Profundidade (seção 3.2) em uma aplicação real também foi atingido. Esta teoria permitiu a extração dos dados do ambiente necessários para a interpretação das regras e geração das sugestões de ações para manter o veículo inteligente em segurança.

O Simulador de Tráfego de Veículos (capítulo 4) foi uma ferramenta muito útil na geração dos picos e na quantificação dos dados nele contidos. Foi utilizado na realização de todos os testes e geração das bases de conhecimento implementadas no Sistema de Segurança em Tráfego de Veículos. O simulador também pode ser considerado em um trabalho futuro, sendo aprimorado para que os carros presentes nas cenas tenham ações de acordo com as sugestões do sistema. Além disso, podem ser acrescentados semáforos e placas de trânsito.

Foi considerado apenas um veículo inteligente interagindo com diversos veículos comuns dirigidos por pessoas habilitadas a tomarem decisões no caso de concorrência ou *dead lock*. Caso exista mais de um veículo inteligente, os axiomas devem ser adaptados para que não haja nenhum tipo de travamento, como por exemplo, dois carros que param e não se movimentam mais por não existir a distância mínima de segurança entre eles. Uma opção para solucionar um caso como este seria o conceito de FILA (*first in first out*).

Até este momento foram utilizadas fórmulas lógicas que buscam na base de conhecimento apenas dados extraídos da visão frontal do agente. O aprimoramento do sistema poderá ser realizado em trabalhos futuros, por exemplo, através do acréscimo da visão dos retrovisores e outros dados de entrada necessários. O mesmo sistema poderá ser utilizado sem problemas, apenas acrescentando-se as devidas fórmulas para interpretar os novos dados e complementar o raciocínio de ações. Sugere-se também a aplicação deste sistema em robôs móveis, ou qualquer outro equipamento capaz de ser influenciado pelas regras propostas, além da instalação deste sistema em um veículo real, o que acarretará em sua adaptação para lidar com incertezas e inserção de leis de

trânsito com um sistema de controle do veículo.

APÊNDICE A – ARTIGOS PUBLICADOS

Artigo publicado durante o mestrado:

Adeline Cecilia Sécolo e Paulo Eduardo Santos. Raciocínio Espacial Qualitativo Para Um Sistema de Segurança em Tráfego de Veículos. SBA Controle & Automação Sociedade Brasileira de Automática, vol.22 2011.

REFERÊNCIAS

- COHN, A. G. et al. Representing and reasoning with qualitative spatial relations about regions. *Spatial and Temporal Reasoning*, p. 97–134, 1997.
- COHN, A. G.; COHN, F.; HOGG, D. C. Building qualitative event models automatically from visual input. In: *ICCV*. [S.l.]: IEEE, Narosa Publishing House, 1998. p. 350–355.
- COHN, A. G.; HAZARIKA, S. M. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, v. 46, n. 1-2, p. 1–29, 2001.
- COHN, A. G. et al. Cognitive vision: Integrating symbolic qualitative representations with computer vision. In: *Cognitive Vision Systems*. [S.l.: s.n.], 2006. p. 221–246.
- FERRARI, F. et al. *Simulação de Tráfego de Veículos por Raciocínio de Ações e Mudanças*. [S.l.], 2008.
- FRANK, T. et al. Characterization of occlusion situations occurring in real-world traffic scenes. In: *Workshop on ECCV*. [S.l.]: Elsevier Science, 1996. p. 43–57.
- GALATA, A. et al. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In: *ECAI*. [S.l.: s.n.], 2002. p. 741–745.
- GERBER, R.; NAGEL, H.-H.; SCHREIBER, H. Deriving textual descriptions of road traffic queues from video sequences. *European Conference on Artificial Intelligence*, p. 736–740, 2002.
- HU, X.; TAO, C. V.; HU, Y. Automatic road extraction from dense urban area by integrated. In: *IAPRSIS*. [S.l.: s.n.], 2004. p. 288–292.
- HUMMEL, B. *Description Logic for Scene Understanding at the Example of Urban Road Intersections*. Tese (Doutorado) — Universität Karlsruhe, 2009.
- KÖHLER, C.; LUDWIGS, A. The occlusion calculus. *Workshop on Cognitive Vision*, 2002.
- LEVESQUE, H.; PIRRI, F.; REITER, R. Foundations for the situation calculus. *Electronic Transactions on Artificial Intelligence*, v. 2, p. 159–178, 1998.
- MCCALL, J. C.; TRIVEDI, M. M. Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE*, v. 7, n. 1, p. 20–37, 2006.
- MERZ, S.; WIRSING, M.; ZAPPE, J. A spatio-temporal logic for the specification and refinement of mobile systems. In: *FASE*. [S.l.: s.n.], 2003. p. 87–101.
- MULLER, P. A qualitative theory of motion based on spatio-temporal primitives. In: *KR*. [S.l.: s.n.], 1998. p. 131–143.
- NAGEL, H.-H. Steps toward a cognitive vision system. *AI Magazine*, v. 25, n. 2, p. 31–50, 2004.
- NARANJO, J. et al. Overtaking maneuver experiments with autonomous vehicles. *ICAR*, p. 1699–1703, 2003.
- PEREIRA, F. C. N.; SHIEBER, S. M. *Prolog and Natural-Language Analysis*. [S.l.]: Microtome Publishing, 2005. 204 p.

- RANDELL, D. A.; CUI, Z.; COHN, A. G. A spatial logic based on regions and connection. In: *KR*. [S.l.: s.n.], 1992. p. 165–176.
- RANDELL, D. A.; WITKOWSKI, M. Building large composition tables via axiomatic theories. In: *KR*. [S.l.: s.n.], 2002. p. 26–36.
- RANDELL, D. A.; WITKOWSKI, M.; SHANAHAN, M. From images to bodies: Modelling and exploiting spatial occlusion and motion parallax. In: *IJCAI*. [S.l.: s.n.], 2001. p. 57–66.
- SANTOS, M. V. et al. Logic-based interpretation of geometrically observable changes occurring in dynamic scenes. *Applied Intelligence*, v. 31, p. 161–179, 2009.
- SANTOS, P. *Spatial Reasoning and Abductive Interpretation of Sensor Data Obtained by a Mobile Robot in a Dynamic Environment*. Tese (Doutorado) — Imperial College, 2003.
- SANTOS, P. Reasoning about depth and motion from an observer’s viewpoint. *Spatial Cognition and Computation*, v. 7, n. 2, p. 133–178, 2007.
- SANTOS, P.; FENELON, V.; DEE, H. M. Knowledge-based adaptive thresholding from qualitative robot localisation using cast shadows. *European Conference on Artificial Intelligence*, p. 1089–1090, 2010.
- SANTOS, P.; SHANAHAN, M. Hypothesising object relations from image transitions. In: *European Conference on Artificial Intelligence*. [S.l.: s.n.], 2002. p. 292–296.
- SOUTCHANSKI, M.; SANTOS, P. Reasoning about dynamic depth profiles. In: *European Conference on Artificial Intelligence*. [S.l.: s.n.], 2008. p. 30–34.
- THRUN, S. A personal account of the development of stanley, the robot that won the darpa grand challenge. *AI Magazine*, v. 27, n. 4, p. 69–82, 2006.
- WIELEMAKER, J. An overview of the swi-prolog programming environment. *Workshop on Logic Programming Environments*, p. 1–16, 2003.
- YAMAN, F.; NAU, D. S.; SUBRAHMANYAN, V. S. A logic of motion. In: *KR*. [S.l.: s.n.], 2004. p. 85–94.
- YAMAN, F.; NAU, D. S.; SUBRAHMANYAN, V. S. Going far, logically. In: *IJCAI*. [S.l.: s.n.], 2005. p. 615–620.