

自動運転システムにおける画像を対象とした 形式仕様記述言語 BBSL の提案

田中 健人¹ 青木 利晃¹ 川上 大介² 千田 伸男² 河井 達治¹ 富田 堯¹

概要: 自動運転システムの安全性評価のため、各国の監督省庁では、走行環境を複数のシナリオで定義し、ユースケースを体系化することで、安全基準を定め、評価の枠組みを提案しているが、この枠組みに基づいて記述される現状の仕様は図や自然言語を用いて曖昧な表現がされている。そこで、本研究では自動運転システムの仕様を厳密に記述する言語、Bounding Box Specification Language(BBSL)を提案する。BBSLは自動運転システムのタスクのうち、「対象物とイベントの検知と反応」に関する仕様を記述するための言語である。特に、BBSLは区間解析の体系を独自に拡張して定義しており、自動運転システムの前方カメラから入力される画像の視点で各オブジェクトを bounding box として記述する。BBSLにより、自動運転システムの多様な走行環境を現実的な分量で、厳密に記述可能となる。BBSLの評価のために行った2種類の実験を示す。1つ目は、自動運転システムの「対象物とイベントの検知と反応」に関する仕様を実際に記述する実験である。この実験によりBBSLの記述能力を示す。2つ目は、BBSLと同様に、画像上のオブジェクト同士の関係の記述に用いられる既存の手法と比較を行う。この実験により表現力や抽象度の違いを述べ、BBSLが他手法より自動運転システムの仕様に優れていることを示す。

キーワード: 形式仕様, 自動運転, 区間, bounding box

1. はじめに

自動運転システムは、次世代の運転技術として期待されており、世界的に開発が進められている [1]。人命に関わり、高い安全性が要求されるため、その安全性評価は最重要課題の一つである。しかし、自動運転システムはオブジェクトの認識のための機械学習 [2] やセンシング技術 [3]、経路探索アルゴリズム [4] など様々な要素からなる非常に複雑なシステムであること、そして、多様な環境での走行が考えられるため、安全性に関する要件を厳密に定義することは容易ではない。そこで、各国の監督省庁では、走行環境を複数のシナリオで定義し、ユースケースを体系化することにより安全基準を作成し、この基準に基づいてシステムを評価する一連の手法を提案している。具体的には、米国の国家道路交通安全局 National Highway Traffic Safety Administration(NHTSA)の研究 [5] やドイツの連邦経済エネルギー省の基金で運営される PEGASUS プロジェクト [6]、日本自動車研究所 (JARI)の研究 [7] などがある。

例えば、NHTSAが研究・提案した手法 [5] では、既存の

自動運転システムの分類や、各自動運転システムで実現すべき機能などを定め、徐々に具体的な機能に落とし込みながら系統的に安全性に関する要件を定める手法が提案されている。

あらゆる状況を詳細に記述することは現実的でないため、このような手法に沿って、システムの仕様を定義する際、どういった走行環境の場合にどのような操作を決定させるかは適切な抽象度で書き下さなければならない。そのため、現状の仕様は図や自然言語を用いて曖昧な表現がされている。例えば、図1は、JARIの研究 [7] で作成された、1ユースケースである。これは、天候などの自然環境や、各車両の速度の初期設定などは、明確に記されているが、自車両と2つの他車両の初期位置は図で表現され、解釈の仕方が複数考えられる曖昧な表現となっている。

仕様は他の開発者と方向性を共有するために用いるものであることから、その解釈の一意性を担保するためには曖昧性をなるべく排除しなければならない。そこで、このような非形式的な図を用いるのではなく、厳密な仕様の作成が求められる。

走行環境上のオブジェクトを厳密に記述できる手法を考えたとき、1つに画像認識システムの評価などに用いられる Intersection over Union(IoU)による記述がある。写真

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
² 三菱電機株式会社
Mitsubishi Electric Corporation

No.	基本ユースケース	基本要件	初期イメージ	初期設定				
				種類	車線	方向	速度 km/h	TTCA(s)
23	23 本線	①通過	直線(2000, 4車線)	白車	直線	100		
		②対象物	1つ (停止車線)	緑車	直線	0	5	130
		③対象物	緑車 (又は赤車)	緑車	直線	100		
		④対象物	緑車 (又は赤車)	緑車	直線	100		

図 1 JARI の研究で示されているユースケース例 [7]



図 2 前方からの写真を IoU で記述する例

上の対象を検知するとき、画像認識システムはその対象のオブジェクトを bounding box で囲って検知する方式が多く使用されており、システムの出力結果を示す際にも図 2 の各写真のように可視化されることが多い。IoU は、2 つの bounding box の全体の領域に対する重なった領域の割合で求められ、画像認識の精度を競い合うコンペ PASCAL VOC[8] などでも採用されている評価の値である。図 2 は、2 枚の写真上に、正解領域を示す赤い bounding box と予測領域を示す緑色の bounding box を描いたものであり、2 つの bounding box の IoU は右の写真でも左の写真でも、共に 1/3 程度となり、同じ値となる。しかし、自動運転システムの仕様を考えたとき、左の写真のズレは直進中の自動運転システムに影響を与えるズレであり、右の写真のズレはどちらにしろ直進中の自動運転システムには問題のないズレといえる。このように、自動運転システムの仕様を記述する際に、重なりのみでの記述は表現力が低く、適していないと考えられる。

一方で、デカルト座標系を用いた幾何学的な記述を行えば、オブジェクト同士の位置関係を厳密に記述することが可能であり、具体的な値を用いて細かい事柄も記述することができる。しかし、具体的な値を用いると、各国の監督官庁・業界団体が提案する安全性評価の枠組みに合った抽象度で仕様を書き下すことができなくなってしまう。

以上の問題点より、具体的な値に依存しない抽象度で、自動運転システムの安全性を記述可能な厳密な言語を開発する必要があった。自動運転における画像認識システムは「対象物とイベントの検知と反応」のタスクの実現に用いられ、全体の安全性に特に影響を与える。そのため、本研究では、図 2 であげたような入力画像として用いられる前方からの画像で表現できる記述に焦点を当てた。そして、対象となるオブジェクトは、それを包含する四角形である bounding box として考えた。本研究の目的は、現状では曖昧な図を用いて表現されている自動運転システムの「対象物とイベントの検知と反応」に関する仕様・検証シナリオのうち、画像上のオブジェクト同士に関する記述を対象に、bounding box 同士の関係として NHTSA の手法と同程度の抽象度で形式的に記述する方法を提案することにある。ここで、画像上のオブジェクトを bounding box で表

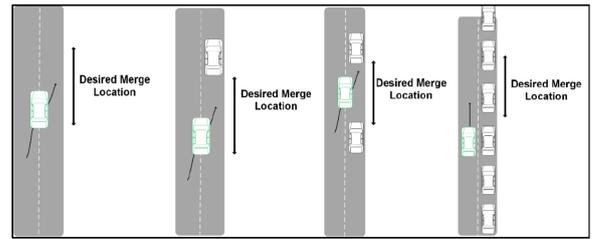


図 3 NHTSA の研究で図示されている合流時の 4 シーン [5]

表 1 BBSL の基本型

real	実数型
bool	ブール値型
interval	区間型
bb	bounding box 型
setBB	bounding box の集合型

現することにしたのは、図 3 のように対象の抽象度ではオブジェクトの詳細な形についての記述はなく、オブジェクト同士の位置関係に着目しており、画像認識システムの入力画像を考えたとき bounding box で記述の方が扱いやすいと考えたためである。

2. BBSL

画像上のオブジェクトを bounding box として記述し、その bounding box 同士の位置関係を NHTSA などが対象としている抽象度で記述できる言語として本研究では Bounding Box Specification Language(BBSL) を提案する。

この研究のアイディアは、bounding box を区間解析 [9] の体系における 2 次元の区間で定義し、この体系では記述できない位置関係について、専用の関係や関数を追加することで、画像上の位置関係を bounding box (あるいは、bounding box の集合) 同士の関係として厳密に記述できるようにしたことにある。

ここでは、初めに BBSL の型、関係、文法を順番に取り上げ、それによりどういう特徴を有しているか紹介する。

2.1 型

BBSL は表 1 のような型を持っている。

BBSL は real 型で実数値の変数や、関数を定義できるが、bounding box 間の距離などは、具体的な値で表現せず、区間や bounding box で抽象化して表現する。そのため、real 型は主に割合などを示す値として利用される。

interval 型は、区間を表現する。この区間は区間解析 [9] で定義されているものと同様で、定義 2.1 となる。画像上の車両などのオブジェクトは後述する bb 型で表現するが、維持すべき車間距離のような対象は interval 型で表現することが可能となる。

定義 2.1. 区間 a とは、 \underline{a}, \bar{a} をそれぞれ区間の端点として以下で定義される。

表 2 BBSL の基本関数

	関係・関数
(1)	$\langle, \rangle, = : \text{interval} \times \text{interval} \rightarrow \text{bool}$
(2)	$\approx : \text{interval} \times \text{interval} \rightarrow \text{bool}$
(3)	$\subset, \supset : \text{interval} \times \text{interval} \rightarrow \text{bool}$
(4)	$\text{PROJ}_i() : \text{bb} \rightarrow \text{interval}$ $\ast i \in \{x, \underline{x}, \bar{x}, y, \underline{y}, \bar{y}\}$
(5)	$\approx : \text{bb} \times \text{bb} \rightarrow \text{bool}$
(6)	$\text{not} : \text{bool} \rightarrow \text{bool}$ $\text{and, or} : \text{bool} \times \text{bool} \rightarrow \text{bool}$
(7)	$\text{RAT} : \text{setBB} \times \text{setBB} \rightarrow \text{real}$
(8)	$\langle, \rangle, = : \text{real} \times \text{real} \rightarrow \text{bool}$
(9)	$w : \text{interval} \rightarrow \text{real}$
(10)	$\cap : \text{bb} \rightarrow \text{bb}$
(11)	$\cap, \cup : \text{setBB} \times \text{setBB} \rightarrow \text{setBB}$

$$a = [\underline{a}, \bar{a}] = \{x \in R: \underline{a} \leq x \leq \bar{a}\}$$

bb 型は, bounding box を表現し, この bounding box は 2次元の区間として区間解析での定義 2.2(この場合, $n=2$) で定義される.

定義 2.2. $a_i, i = 1, \dots, n$ を区間とすると, 多次元区間 a を以下のように定義される.

$$a = (a_1, \dots, a_n)$$

また, BBSL はオブジェクトを bounding box の集合で記述可能とするため, setBB 型により, bounding box の集合を表現する.

2.2 関係

BBSL は表 2 のような大きく 11 種類に大別される関係・関数を持っている. 区間の大小関係などは, 区間解析で用いられている関係を使っている. しかし, 区間解析での区間同士の四則演算は誤差付きの実数値を計算するために定義されたものであるため, 本研究の目的の位置関係の記述には応用できない. そこで BBSL では, 画像仕様記述に有用な独自の演算および関係が用意されている. 以下に, 表 2 の各演算について説明する.

(1) の関係は, 2つの区間同士の大きさによって, 位置関係を記述するために用いる基本の関係として導入している. 定義は, 区間解析で用いられるものと同様で, 定義 2.3, 定義 2.4 で与えられる.

定義 2.3. 区間 a, b について, 区間上の二項関係 $<$ は, 以下で定義される.

$$a < b \Leftrightarrow \bar{a} < \underline{b}$$

定義 2.4. 区間 a, b について, 同値関係は以下で定義される.

$$a = b \Leftrightarrow \underline{a} = \underline{b} \text{ かつ } \bar{a} = \bar{b}$$

この関係により, 図 4 にあげるような関係を記述できる. ここで比較可能な 2つの区間は, 同じ次元上の区間であり,



図 4 関係 (1) の例

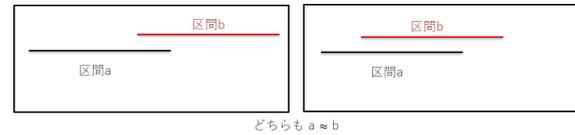


図 5 関係 (2) の例



図 6 関係 (3) の例

並行に位置している 2つの区間が完全に重なっているか, 大きい位置にあるか, 小さい位置にあるかの記述に用いる.

(2) この関係は, (1) の $=$ のように, 完全に一致している場合でなく, $<$, $>$ のように, 完全に離れている場合でもなく, 少しでも重なっている関係を表現するために導入している.

定義は, 以下のように本研究で新たに与えた.

定義 2.5. a, b を区間とすると, 区間上の関係 \approx は以下のように定義される.

$$a \approx b \Leftrightarrow a \cap b \neq \emptyset$$

この関係により, 図 5 にあげるような関係を記述できる. このように, 二つの区間が重なっている場合は, その程度に関わらずどちらも \approx で記述される.

(3) この関係は, (2) のような重なっている場合のうち, 包含関係が成り立っているかを表現するために導入している.

定義は, 区間解析で用いられるものと同様で, 定義 2.6 で与えられる.

定義 2.6. 区間 a, b について, 区間上の二項関係 \subseteq は, 以下で定義される.

$$a \subseteq b \Leftrightarrow \underline{b} \leq \underline{a} \text{ かつ } \bar{a} \leq \bar{b}$$

この関係により, 図 6 にあげるような関係を記述できる. このように, 二つの区間がどちらかに内包していることを記述するとき用いる.

(4) 本研究の, 主な対象は二次元の区間である bounding box であるが, これまで紹介した関係は一次元の区間同士の関係である. bounding box では, x 軸方向, y 軸方向の 2つの区間で定義するため, この関数は bounding box か

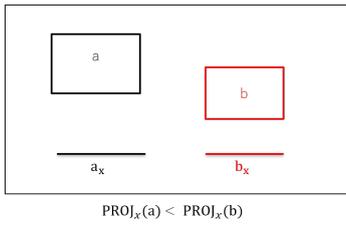


図 7 関数 (4) の例

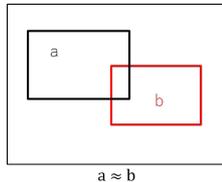
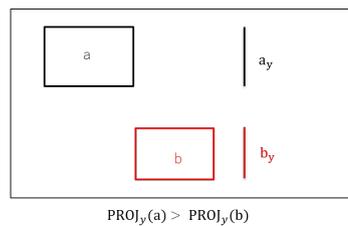


図 8 関係 (5) の例

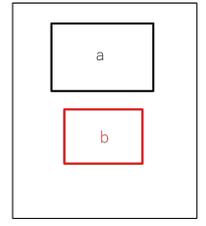


図 9 論理演算 (6) の例

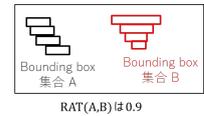


図 10 関数 (7) の例

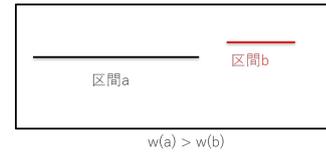


図 11 関数 (9) の例

ら、それぞれの区間に分けるための、サブセクション関数として導入している。

定義は、以下のように本研究で新たに与えた。

定義 2.7. bounding box $a=(a_x, a_y)$ について、区間上の演算 $PROJ_i$ は以下のように定義される。ここで、 $i \in \{x, y\}$ とする。

$$PROJ_x(a) = a_x, PROJ_y(a) = [a_y, a_y], PROJ_x(a) = [a_x, a_x]$$

例えば、二つの bounding box a, b があった場合、(1) のような区間同士の関係を用いて、図 7 にあげるような関係を記述できる。このように、bounding box に関する x 軸、 y 軸についての大小関係により、bounding box b が a より右に位置していることや、 a が b より上に位置していることを記述できる。

(5) この関係は、bounding box の x 軸にあたる区間、 y 軸にあたる区間どちらにも関係 \approx が成り立つことを表現する。このような関係を記述したい場合は多くあることから、これらを簡略化して表現するために、bounding box 同士についても関係 \approx を導入している。

定義は、以下のように本研究で新たに与える。

定義 2.8. bounding box $a=(a_x, a_y), b=(b_x, b_y)$ について、2次元の区間 (bounding box) 上の関係 \approx は以下のように定義される。

$$a \approx b \Leftrightarrow a_x \approx b_x \text{ かつ } a_y \approx b_y$$

となる。

例えば、2つの bounding box a と b が少しでも重なっていることを、図 8 にあげるような関係を記述できる。

(6) これまで定義した関係を複数組み合わせることで記述可能なように、論理否定、論理積、論理和に関する論理演算を導入している。

例えば、bounding box a と b について、 x 軸に関しては a は b を包含しており、かつ、 y 軸に関しては a は b よ

り大きいような関係は、図 9 にあげるような関係を記述できる。

(7) 本研究では車線などの画像上の領域を bounding box の集合で記述する。そのため、この関数は、ある 2つの領域の大きさの比率の記述のために導入している。

定義は、以下のように本研究で新たに与えた。

定義 2.9. Bounding box 集合 A, B とすると、関数 RAT は以下のように定義される。

$$RAT(A, B) = (A \text{ が占める総面積}) / (B \text{ が占める総面積})$$

となる。

例えば、領域を被覆した bounding box 集合 A と B について、この 2つの大きさの比率は 0.9 対 1.0 だとすると、図 10 にあげるような関係を記述できる。

(8) 関数 (7) で計算した比率の大小関係や、関数 (9) で取り上げる区間の幅の大小を表現するために、実数の大小関係も導入している。

例えば、図 9 のような領域を被覆した bounding box 集合 A と B について、 A より B の方が大きいことを表すには、 $RAT(A, B) < 1$ と記述できる。

(9) 各区間の長さについての記述のために、区間の幅を返す関数を導入している。

定義は、区間解析で用いられるものと同様で、定義 2.10 で与えられる。

定義 2.10. 区間 a の幅 $w(a)$ は、以下で定義される。

$$w(a) = \bar{a} - \underline{a}$$

例えば、二つの区間 a, b について、 a の幅が b の幅より大きいことを表現する場合、図 11 のように (8) の大小関係と併用して記述する。

(10) bounding box 同士の重なっている部分を表現するために、bounding box に関する共通部分を計算する 2 項

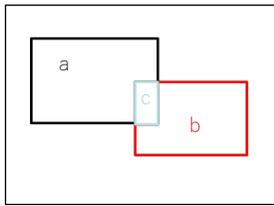
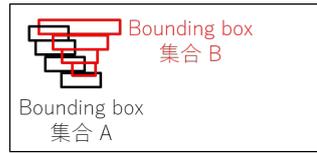


図 12 演算子 (10) の例



$$RAT(A \cap B, A \cup B) < 0.5$$

図 13 関係 (11) の例

演算子を導入している。

定義は、区間解析で用いられるものと同じで、定義 2.11 で与えられる。

定義 2.11. 多次元区間 $a=(a_1, \dots, a_n), b=(b_1, \dots, b_n)$ とすると

$$a \cap b = (a_1 \cap b_1, \dots, a_n \cap b_n)$$

となる。

例えば、bounding box a と b について、この 2 つの重なった bounding box を抽出したい場合、図 12 にあげるような関係を記述できる。

(11) ある領域同士の重なって部分や、全体に対する割合などを表現するために、bounding box の集合に関する共通部分を計算する 2 項演算子を導入している。

これらの定義は、 \cup は一般に用いられる集合の演算子と同じものだが、 \cap は、以下のように本研究で新たに与える。

定義 2.12. bounding box 集合 A, B とすると、 $A \cap B$ は以下のように定義される。

$$A \cap B = \{a \cap b | a \in A, b \in B\} \setminus \emptyset$$

となる。

例えば、領域を被覆した bounding box 集合 A と B について、この 2 つの全体領域に対する重なった領域の割合が 0.5 以下である関係を表現したい場合、図 13 にあげるような関係を記述できる。

2.3 BBSL による記述

BBSL の構文は BNF によって定義した。図 14 は、各構文の規則を定義したものであり、図 15 は、各トークンについて定義したものとなる。BBSL で仕様を記述した例を図 16 に示す。この仕様は、車両 A の位置による、停止すべきケースと停止しないケースを書き分けたものである。

BBSL での仕様は図 16 のように、3 つのブロックに分けられる。

exfunction-endexfunction 要素 (外部関数ブロック) は、この仕様内で記述することと、それ以外の部分を明確に区別することを目的としている。対象のオブジェクトを囲った bounding box を受け取るための関数や、存在をチェックするための関数など、仕様の記述に必要となり、事前に

```

image-spec ::= external-function scenario-condition case+
external-function ::= block-open function-definition * block-close
function-definition ::= function-name "(" type ")" ":" type
scenario-condition ::= block-open "[" condition-definition "]" block-close
condition-definition ::= "(" (" condition-definition condition-binop condition-definition ")")
                    | "(" condition-monop "(" condition-definition ")" )
                    | "(" quantification-definition "(" condition-definition ")" )
                    | condition-term none-token
condition-term ::= value condition-com value
quantification-definition ::= quantification-expr "(" quantification-definition ")" * ":"
quantification-expr ::= quantification-token var-name quantification-token value
value ::= var-name number | function-call
                    | "(" value binop value ) | "(" monop value )
block-open ::= "exfunction" | "condition" | "case"
block-close ::= "endexfunction" | "endcondition" | "endcase"
case ::= block-open case-name case-definition block-close
case-definition ::= let-definition condition-definition
let-definition ::= let-token let-expr "(" let-expr )" * let-token
let-expr ::= var-name ":" type "=" value
function-call ::= function-name [proj-index] "(" value*" ")"
    
```

図 14 BBSL の構文

```

quantification-token ::= ("exists" | "forall") | "∈"
function-name ::= [任意の文字列] | "PROJ" | "RAT" | "w"
var-name ::= [任意の文字列]
type ::= "real" | "bool" | "interval" | "bb" | "setBB"
let-token ::= "let" | "in"
none-token ::= "none"
condition-monop ::= "not"
condition-binop ::= "and" | "or"
condition-com ::= "=" | "<" | ">" | ">=" | "<=" | ">" | "<" | "≥" | "≤"
monop ::= "∩" | "∪"
binop ::= "∩" | "∪"
proj-index ::= "x" | "y" | "z" | "g" | "y" | "g"
number ::= [任意の数値]
    
```

図 15 BBSL のトークン

```

1 exfunction
2 //車両 A の存在をチェック. あれば true を返す
3 車両 A がある ():bool
4 //車両 A を囲む boundingbox を返す
5 車両 A ():bb
6 //減速しなければならない範囲の interval を返す
7 減速区間 ():interval
8 exfunction
9
10 condition
11 [前方車両がある()] } 前提条件
12 endcondition } ブロック
13
14 case 停止
15 let 車両 A:bb = 前方車両(),
16 減速区間:interval = 減速区間 () in
17 PROJy(車両 A) ≈ 減速区間 } ケース
18 endcase } ブロック
19
20 case 停止しない
21 let 車両 A:bb = 前方車両(),
22 減速区間:interval = 減速区間 () in
23 PROJy(車両 A) > 減速区間
24 endcase
    
```

図 16 BBSL による記述例

用意すべき関数を記述する。

condition-endcondition 要素 (前提条件ブロック) は、この仕様を想定している前提条件の記述を行うために用意している。この例では、今回車両 A が必ず存在することを前提としているため、その内容を記述している。

case-endcase 要素 (ケースブロック) には、bounding box 同士の関係によって条件を記述し、複数のケースを繰り返して記述する。仕様の記述のメインとなり、「対象物とイベ

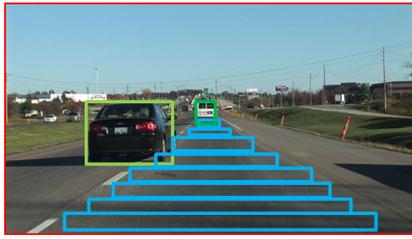


図 17 BBSL で記述するオブジェクト例

「対象物の検知と反応」に関する仕様を記述する際に、対応する反応毎に case を定義し、各条件の記述を行う。let 文により、対象となるオブジェクトなどの値を用意し、図 16 の 17 行目や 23 行目のように後続して条件文を記述する。

2.4 BBSL の特徴

BBSL は前節までに挙げた型、関係、構文を持つことにより、以下の代表的な特徴を有する。

- exfunction-endexfunction 要素 (外部関数ブロック) で外部関数を定義することで、仕様を設計する際、前方からの画像で表現できる仕様と、それ以外の仕様を明確に区別できる。
- bounding box 集合により、図 17 の車線のようなオブジェクトも (ここでは、青い setBB により) 記述することができる。
- Interval や bounding box を用いることで、実数値に依存しない抽象度で距離を記述することができる。
- bounding box 集合に加え、割合が記述可能であることにより、動的なオブジェクトの予測領域との割合で危険な度合いなどを記述できる。
- PROJ 関数により、x 軸、y 軸別々に位置関係を記述することができ、図 18 のように多様な関係を記述することができる。

3. 実験と評価

BBSL を評価するため、2 種類の実験を行った。1 つ目は、NHTSA の手法に則り、実際の自動運転システムの仕様のうち「対象物とイベントの検知と反応」に関する仕様を反応ごとに書き分けて記述することで、目的とした抽象度での BBSL の記述能力を評価する実験である。2 つ目は、画像上のオブジェクト同士の関係を記述できる既存の手法を用いた記述と BBSL を用いた記述を比較し、既存研究との差分について明らかにする実験である。

3.1 NHTSA の仕様作成方法に則った記述実験

この実験では、NHTSA が研究 [5] で提案した自動運転システムの設計・テストフレームワークのうち、「対象物とイベントの検知と反応」に関する仕様を BBSL を用いて記述した。NHTSA の研究で、どういうイベントが存在し、その時どういうレスポンスが必要か示されており、その全

表 3 L3TJD についての記述結果

対象のイベント	行数	記述したレスポンス数
前方車両が減速している	42	4
前方車両が止まっている	30	3
前方車両が加速している	30	3
前方車両が旋回している	38	3
他車がレーン変更している	51	3
他車が自車レーンに割り込んでいる	56	4
他車が道路に侵入している	56	4
隣接車両が自車線に侵入している	56	4
対向車両が自車線に侵入している	67	5
前方車両が自車線から外にカットアウトしている	45	4
自転車が道路を横断している (横断歩道内)	64	4
自転車が道路を横断している (横断歩道外)	64	4
自転車が自線上にいる	34	2
自転車が専用車線上にいる	26	2
歩行者が道路を横断している (横断歩道内)	64	4
歩行者が道路を横断している (横断歩道外)	64	4
車線上に静的な障害物がある	44	3
動的な障害物がある	46	3
合計	877	63

てのイベントに対して、各レスポンスを前方写真からの記述で BBSL を用いて書き分けた。記述した具体的なイベントを表 3 に示す。

3.2 既存の記述方法との比較実験

この実験では、BBSL と、画像上のオブジェクトの記述に用いられる既存の手法とで比較を行い、差異について評価した。画像上のオブジェクト同士の関係を記述できる手法としては、提案手法である BBSL 以外にも、画像認識システムの評価指標としてよく用いられる IoU、地図データベースの記述などに用いられる位相的な記述 binary topological relationships(btr)[10]、より具体的なアルゴリズムの設計に用いられる幾何学的な記述 [11] などがあげられる。

IoU、幾何学的な記述と比較した結果、それぞれ表 4、表 5 となった。また、表 6 は、位相的な記述である btr と BBSL を比較した結果であり、アルファベットで記述した各関係の名称は btr で定義されているものに準拠している。

3.3 評価・考察

表 3 が示すように、BBSL は NHTSA の研究で示唆された各イベントに対して、区別すべきレスポンスを書き分け

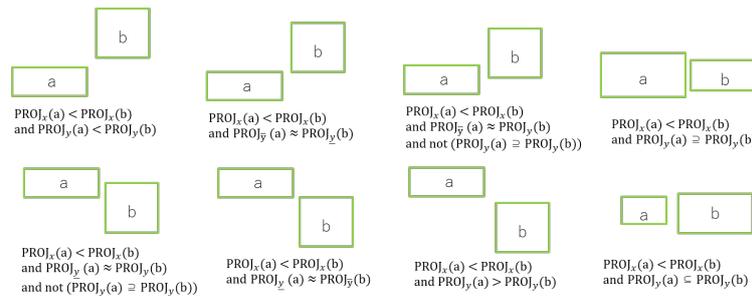


図 18 BBSL で表現できる位置関係の例

表 4 IoU との比較

	提案手法での記述	IoU での記述
割合の表現	表現できる	表現できる
位置関係の表現	表現できる	表現できない
包含関係の表現	表現できる	表現できる
大小関係の表現	表現できる	表現できる
どの程度離れているかの表現	表現できる	表現できない
X 軸, Y 軸への分割	分割できる	分割できない
物体検出システムの評価に対する汎用性	同程度	同程度
計算の簡易さ	簡易的でない	簡易的

表 5 幾何学的な記述との比較

	提案手法での記述	幾何学的な記述
具体的な値を必要とする記述	適切でない	適切
具体的な値を排除した記述	適切	適切でない

表 6 btr との比較

	提案手法での記述	btr での記述
A contains B	表現できる	表現できる
A covers B	表現できる	表現できる
A touch B	表現できる	表現できる
A overlapbdyintersect B	表現できる	表現できる
A overlapbdydisjoint B	表現できる	表現できる
A equal B	表現できる	表現できる
A disjoint B	表現できる	表現できる
B on A	表現できる	表現できる
割合	表現できる	表現できない
距離	表現できる	表現できない
大小関係	表現できる	表現できない
多次元空間への応用	確立されていない	確立されている

る表現力を有していることが分かった。そして、具体的な距離や自車の速度に依存しない記述が可能であることを確認できた。しかし、実際に記述を行うと、オブジェクトの数やレスポンス数が多くなるほど、BBSL で記述した仕様の分量も多くなり、冗長な表現が増えた。これは、仕様の可読

性を損ない、誤認する要因となりうるため、冗長な表現を避ける改良が必要である。

既存手法の比較からは、表 4 が示すように、IoU では表現できない位置関係や、どの程度離れているかの距離について、BBSL では記述ができることを確認できた。しかし、大量のデータに対し、IoU を計算する場合、単純なプログラムで比較的高速に計算できることに対し、BBSL による記述の各条件を計算する場合、BBSL で書かれた仕様を処理できる専用のツールの作成が必要となり、その計算は IoU より複雑で、時間がかかることが予測できる。また、表 5 からは、抽象度についての差分を示せた。つまり、幾何学的な記述と、BBSL による記述では対象としている抽象度が明確に異なり、BBSL による記述は、幾何学的な記述より、具体的な値を排除した抽象度で記述することを得意としている。そして、表 6 が示すように、btr で定義された 8 種類の関係のうち、その全てが BBSL でも記述できたことを確認した。また、BBSL と異なり、btr では割合、距離、大小関係といった関係を記述することができない。しかし、3 次元空間上のオブジェクトの記述に应用することを考えたとき、btr は多次元空間上の関係として定義されており、実用例もあることに対し、BBSL で定義した関係は、2 次元の画像上の関係のみを想定しており、3 次元空間への応用方法は確立されていない。本研究の対象は自動運転における画像認識システムの入力画像に焦点をあてているが、3 次元空間上の記述を考える場合、この 2 つの対応について議論が必要となることが予測できる。

4. 関連研究

システムの大規模化・複雑化が進む中、ソフトウェアの品質を確保するため、どのように仕様を記述するかは重要なテーマの 1 つとされている。特に自然言語などを用いた曖昧な表現を含む仕様は、複数の意味に解釈できる記述を含んだり、記述漏れを検出しにくかったりする点で問題視されている。そのため、形式仕様記述言語の重要性が高まっている。代表的なものとして、VDM[12] や Z 言語 [13] などがある。

しかし、これらの形式仕様記述言語は、実数値の大小関係を用いた記述や、集合的な記述によるデータの状態の表

現は行えるが、画像や動画のような複雑で大量の入力に関する取扱は考慮されておらず、それらに関する仕様の記述は困難である。自動運転システムの仕様を抽象的に扱うために、大量の画像を抽象化して捉える方法として、本研究では区間解析 [9] の体系を拡張した。この体系は、元来誤差付きの実数値を計算するために定義されたものであり、位置関係を取り扱うためには拡張する必要がある。

画像に関する仕様を記述する試みとして、地図データベースの記述などに用いられる位相による記述がある [10]。この研究では、地図情報や、CAD などの空間情報の処理への応用のため、1 次元上の位相による位置関係を多次元に適応可能なように拡張して定義している。また、絵画の検索への応用として、画像表現のモデル化に関する研究がある [14]。この研究では、人間の思考にあった検索が可能な絵画検索システムのために、画像情報を記述し、検索をかける手法を提案している。このとき、画像上のオブジェクトの位置関係を、各オブジェクトを表現するアイコンの最小外接長方形 (オブジェクトを内包する最小の bounding box) の重心を参照点として、事前に決められた 12 方向に位置するオブジェクトに対しての位置関係をグラフで記述している。これらの研究は、いずれも画像上のオブジェクト同士の厳密な記述を行っているが、特定分野における画像を対象としたものである。3 節に示したように、本研究で提案した BBSL は bounding box に関する位置関係を詳細に記述できることから、既存研究よりも自動運転システムの安全性の仕様 [5] を画像を基に記述することに適している。

5. まとめ・今後の課題

BBSL を用いることで、自動運転システムの安全性の仕様について、画像上のオブジェクト同士の関係を NHTSA が提案した手法の抽象度で、形式的に記述することができた。現状、これらの仕様は非形式的に表現されているため、曖昧性があり問題があったが、本研究によりこれらの仕様を厳密に定義し、共有・議論することが可能となったといえる。そして、BBSL は既存の手法である位相的な位置関係を記述する方法 [10] や IoU では記述できなかった位置関係を記述できるようになり、幾何学的な記述のように具体的な値に依存した記述を避けることができた。これらの理由から、自動運転システムの「対象物とイベントの検知と反応」に関する安全性の仕様を記述するという目的には BBSL は重要で、新規性がある研究となった。

今後の課題について 2 つ述べる。1 つ目は、本研究の手法を動画上のオブジェクト同士の関係をより詳細に記述できるように拡張することである。提案手法は、システムが想定する走行シーンを静止画像上のオブジェクト同士の関係として記述するものであり、動的なオブジェクトは外部関数で bounding box 集合として受け取り関係を記述して

いる。これを動画のオブジェクトとしてどちらの方向に進んでいるかなどの情報を記述することができれば、表現力が上がり記述できるシナリオが飛躍的に増加する。

2 つ目は、本研究で提案した手法により、記述した仕様の正しさを自動的に検出するシステムを構築することである。提案手法は区間解析の理論をベースに自動運転システムの安全性の仕様を形式的に記述することを可能とした。そのため、計算機により実行し、自動的に仕様を検証できるようなツールへの応用が期待できる。

参考文献

- [1] 青木啓二. 自動運転車の開発動向と技術課題: 2020 年の自動化実現を目指して. 情報管理, Vol. 60, No. 4, pp. 229–239, 2017.
- [2] Aysegül Uçar, Yakup Demir, and Cüneyt Güzeliş. Object recognition and detection with deep learning for autonomous driving applications. *Simulation*, Vol. 93, No. 9, pp. 759–769, 2017.
- [3] Christian Waldschmidt and Holger Meinel. Future trends and directions in radar concerning the application for autonomous driving. In *2014 11th European Radar Conference*, pp. 416–419. IEEE, 2014.
- [4] Xiaohui Li, Zhenping Sun, Dongpu Cao, Zhen He, and Qi Zhu. Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. *IEEE/ASME Transactions on Mechatronics*, Vol. 21, No. 2, pp. 740–753, 2015.
- [5] Eric Thorn, Shawn C Kimmel, Michelle Chaka, Booz Allen Hamilton, et al. A framework for automated driving system testable cases and scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety Administration, 2018.
- [6] Research project PEGASUS. The pegasus method. <https://www.pegasusprojekt.de/en/pegasus-method>.
- [7] 中村英夫, 金子貴信. 自動運転システム安全設計—第 2 報: ユースケース及び機能レベル基本アーキテクチャの研究—. *JARI Research Journal*, 2016.
- [8] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, Vol. 111, No. 1, pp. 98–136, 2015.
- [9] Ramon E Moore, R Baker Kearfott, and Michael J Cloud. *Introduction to interval analysis*, Vol. 110. Siam, 2009.
- [10] Max J Egenhofer. A formal definition of binary topological relationships. In *International conference on foundations of data organization and algorithms*, pp. 457–472. Springer, 1989.
- [11] 北脇卓, 呉海元, 塩山忠義. 画像による横断歩道の距離推定. 計測自動制御学会論文誌, Vol. 38, No. 5, pp. 495–500, 2002.
- [12] Cliff B Jones. *Systematic software development using VDM*, Vol. 2. Prentice Hall Englewood Cliffs, 1990.
- [13] J Michael Spivey and JR Abrial. *The Z notation*. Prentice Hall Hemel Hempstead, 1992.
- [14] 西山晴彦, 松下温ほか. 画像の構図を用いた絵画検索システム. 情報処理学会論文誌, Vol. 37, No. 1, pp. 101–109, 1996.