

〔非公開〕

TR-C-0038

図形画記述／検索 SPADE SYSTEM

ルール説明書

島 則之
NORIYUKI SHIMA

高橋 友一
TOMOICHI TAKAHASHI

1990. 1. 10

A T R 通信システム研究所

図形画記述 / 検索 SPADE SYSTEM
ルール説明書

島 則之 高橋友一
(株)ATR通信システム研究所
知能処理研究室

1月10日, 1990

目次

1	はじめに	4
1.1	概要	4
1.2	ルールの説明	4
1.3	ディレクトリ構成	4
2	LEVEL0 のルール	6
3	LEVEL1 のルール	6
4	LEVEL2 のルール	7
4.1	中央 (空間指示 smiddle)	7
4.2	上 (空間指示 stop)	7
4.3	下 (空間指示 sbottom)	8
4.4	右 (空間指示 sright)	8
4.5	左 (空間指示 sleft)	9
4.6	右上 (空間指示 stop-sright)	9
4.7	左上 (空間指示 stop-sleft)	10
4.8	右下 (空間指示 sbottom-sright)	10
4.9	左下 (空間指示 sbottom-sleft)	11
4.10	横端 (空間指示 edge-x)	11
4.11	縦端 (空間指示 edge-y)	12
4.12	近く (相対指示 near) / 遠く (相対指示 far)	12
4.13	上 (相対指示 up)	13
4.14	下 (相対指示 down)	15
4.15	右 (相対指示 right)	17
4.16	左 (相対指示 left)	19
4.17	重なる (相対指示 overlap)	21
4.18	隣 (相対指示 neighbor)	21
4.19	ダミー (dummy2)	22
5	LEVEL3 のルール	23
5.1	一般的なルール	23
5.2	端 (空間指示 S 端)	23
5.3	隅 (空間指示 S 隅)	23
5.4	横 (相対指示 横)	24
5.5	ダミー	24
6	shi-cf-calc.lisp について	25
6.1	cf-stop	25
6.2	cf-sbottom	25
6.3	cf-sright	25
6.4	cf-sleft	25
6.5	cf-smiddle	25
6.6	cf-stop-sright	25
6.7	cf-stop-sleft	25
6.8	cf-sbottom-sright	25
6.9	cf-sbottom-sleft	25
6.10	val-edge	25
6.11	hashi-calc	25
6.12	sumi-calc	25

6.13	cf-up	26
6.14	cf-down	26
6.15	cf-right	26
6.16	cf-left	26
6.17	likesup	26
6.18	likesdown	26
6.19	likesright	26
6.20	likesleft	26
6.21	val-neighbor	26
6.22	test-cross	26
6.23	test-position	26
A	付録 『上らしさ』の計算方法	27
B	付録 『下らしさ』の計算方法	28
C	付録 『右らしさ』の計算方法	29
D	付録 『左らしさ』の計算方法	30

1 はじめに

1.1 概要

本説明書では、図形画記述 / 検索システムである SPADE SYSTEM (SPAtial-relationship based DEscription/retrieval SYSTEM) における位置関係のルール記述のうち、新たに定義を行なったルールに関する説明を行なう。 SPADE SYSTEM のルール記述の詳細については、東洋情報システム (株) 納入の図形画記述プログラムシステム説明書を参照して下さい。

1.2 ルールの説明

SPADE のルールは、0 から 4 の 5 段階の LEVEL から成り立っている。総数で、約 100 のルールがあるが、このうちオリジナルから変更をしたルールについてのみ説明をする。

1.3 ディレクトリ構成

新たに定義を行なったルールに関連するファイル構成は以下の通りである。

1. CS08:/usr1/flep/shigeru/spade/level0.lisp

- (a) LEVEL0 のルールを記述したファイル。
- (b) area, center等の関係を定義している。

2. CS08:/usr1/flep/shigeru/spade/level1.lisp

- (a) LEVEL1 のルールを記述したファイル。
- (b) distance, azimuth等の関係を定義している。

3. CS08:/usr1/flep/shigeru/spade/level2.lisp

- (a) LEVEL2 のルールを記述したファイル。
- (b) right, up等の関係を定義している。

4. CS08:/usr1/flep/shigeru/spade/level3.lisp

- (a) LEVEL3 のルールを記述したファイル。
- (b) 『上』, 『右』等の関係を定義している。

5. CS08:/usr1/flep/shigeru/spade/likeup.4.02.o

- (a) LEVEL2 の相対指示の up の cf 値を計算する関数を定義している。
- (b) shi-cf-calc.lisp ファイルの Lisp 関数が Call している。

6. CS08:/usr1/flep/shigeru/spade/likedown.4.02.o

- (a) LEVEL2 の相対指示の down の cf 値を計算する関数を定義している。
- (b) shi-cf-calc.lisp ファイルの Lisp 関数が Call している。

7. CS08:/usr1/flep/shigeru/spade/likeright.5.02.o

- (a) LEVEL2 の相対指示の right の cf 値を計算する関数を定義している.
- (b) shi-cf-calc.lisp ファイルの Lisp 関数が Call している.

8. CS08:/usr1/flep/shigeru/spade/likeleft.4.02.o

- (a) LEVEL2 の相対指示の left の cf 値を計算する関数を定義している.
- (b) shi-cf-calc.lisp ファイルの Lisp 関数が Call している.

9. CS08:/usr1/flep/shigeru/spade/neighbor.1.00.o

- (a) LEVEL2 の相対指示の neighbor の cf 値を計算する関数を定義している.
- (b) shi-cf-calc.lisp ファイルの Lisp 関数が Call している.

10. CS08:/usr1/flep/shigeru/spade/cross.o

- (a) LEVEL2 の相対指示の up, down, right, left ルールの条件チェックの関数を定義している.
- (b) shi-cf-calc.lisp ファイルの Lisp 関数が Call している.

11. CS08:/usr1/flep/shigeru/spade/position.o

- (a) LEVEL2 の相対指示の up, down, right, left ルールの条件チェックの関数を定義している.
- (b) shi-cf-calc.lisp ファイルの Lisp 関数が Call している.

12. CS08:/usr1/flep/shigeru/spade/shi-cf-calc.lisp

- (a) LEVEL2 のルールの cf 値を計算する Lisp 関数を定義している.
- (b) level2.lisp ファイルの Lisp 関数が Call している.

2 LEVEL0 のルール

1. level0.lisp 内で定義している.
2. このレベルのルールの殆んどは変更なし.
3. CENTER-RULE, P-BOUND-RULE で求められた
(rel 0 center) (rel 0 pbound)
という結果はCS08 : /usr1/flep/shigeru/spade/work2というファイルへ append される.
4. 出力フォーマットは図形画記述プログラムシステム説明書 1.3 図形構造情報・意味情報のデータの形態と同じである.

3 LEVEL1 のルール

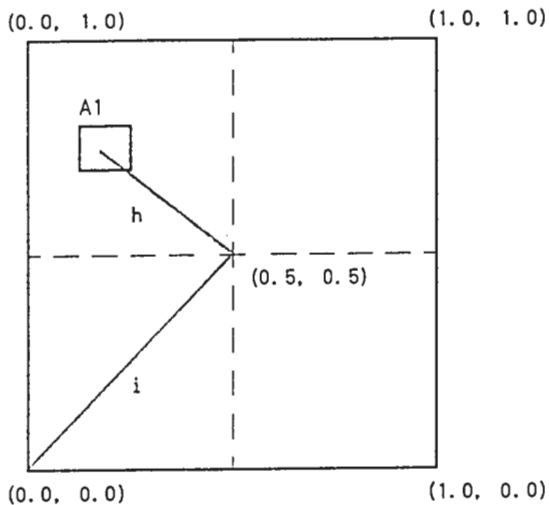
1. level1.lisp 内で定義している.
2. このレベルのルールの殆んどは変更なし.
3. DISTANCE2-RULE で求められた
(rel 1 distance2)
という結果はCS08 : /usr1/flep/shigeru/spade/work2というファイルへ append される.
4. 出力フォーマットは図形画記述プログラムシステム説明書 1.3 図形構造情報・意味情報のデータの形態と同じである.

4 LEVEL2 のルール

- level2.lisp 内で定義している.
- 以下の各ルールの図にでてくる図形 A1 の中心座標を $(x1, y1)$ とする.
- 以下の各ルールの図にでてくる図形 A2 の中心座標を $(x2, y2)$ とする.
- LEVEL2 の RULE の FACT 表現は以下の通りです.
(rel 2 rel-name (objects) (cf 値) 1.0)
- 上記 cf 値として各ルールで定義されている val の値を与える.

4.1 中央 (空間指示 smiddle)

『中央』の対象となる図形の条件チェックはない.



$$h = \sqrt{(x1 - 0.5) * (x1 - 0.5) + (y1 - 0.5) * (y1 - 0.5)}$$

$$i = \sqrt{0.5 * 0.5 + 0.5 * 0.5}$$

$$m = h / i$$

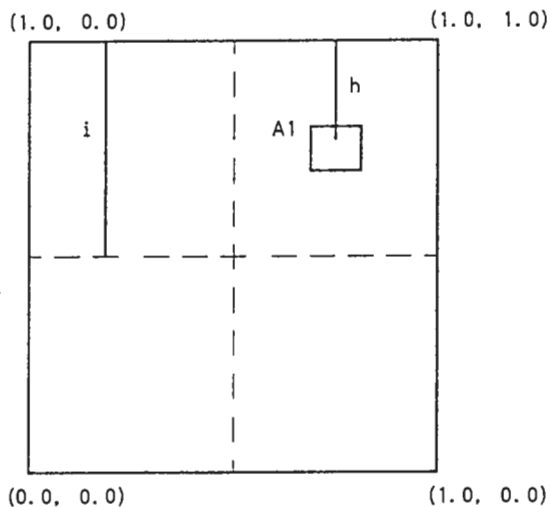
$$val = (1.0 - m) * \exp(-a * m)$$

a はチューニングの為のパラメータです.

計算に必要な h, i の関係図

4.2 上 (空間指示 stop)

『上』の対象となる図形の条件は, 図形の下端が Frame の中心より上にある物のみ.



$$0 \leq \text{azimuth5}(\text{Frame A1}) \leq \pi \text{ の時}$$

$$h = (1.0 - y1)$$

$$i = 0.5$$

$$m = h / i$$

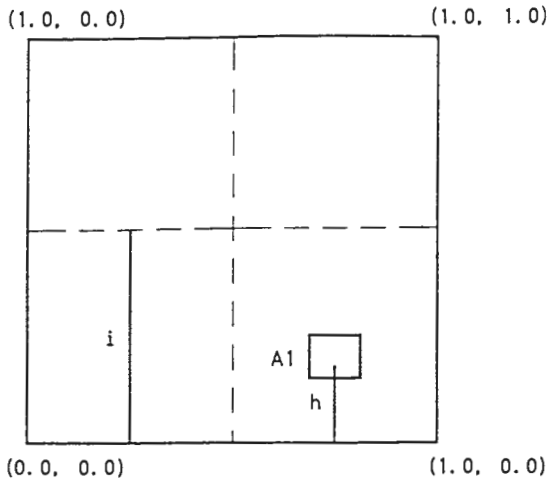
$$val = (1.0 - m) * \exp(-a * m)$$

a はチューニングの為のパラメータです.

計算に必要な h, i の関係図

4.3 下 (空間指示 sbottom)

『下』の対象となる図形の条件は, 図形の上端が Frame の中心より下にある物のみ.



計算に必要な h, i の関係図

$-\pi \leq \text{azimuth4}(\text{Frame A1}) \leq 0$ の時

$$h = y1$$

$$i = 0.5$$

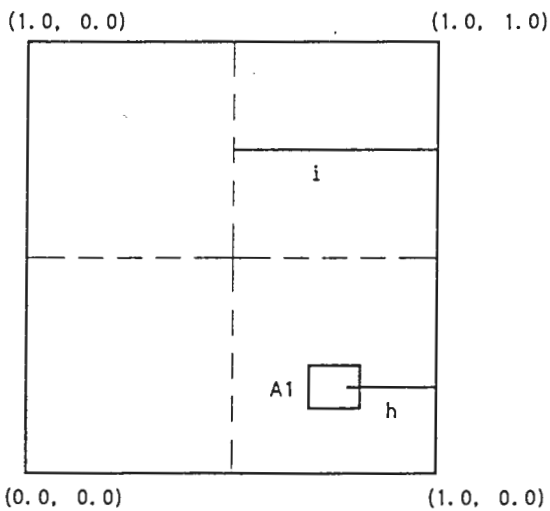
$$m = h / i$$

$$\text{val} = (1.0 - m) * \exp(-a * m)$$

a はチューニングの為のパラメータです.

4.4 右 (空間指示 sright)

『右』の対象となる図形の条件は, 図形の左端が Frame の中心より右にある物のみ.



計算に必要な h, i の関係図

$-\pi / 2 \leq \text{azimuth3}(\text{Frame A1}) \leq \pi / 2$ の時

$$h = 1.0 - x1$$

$$i = 0.5$$

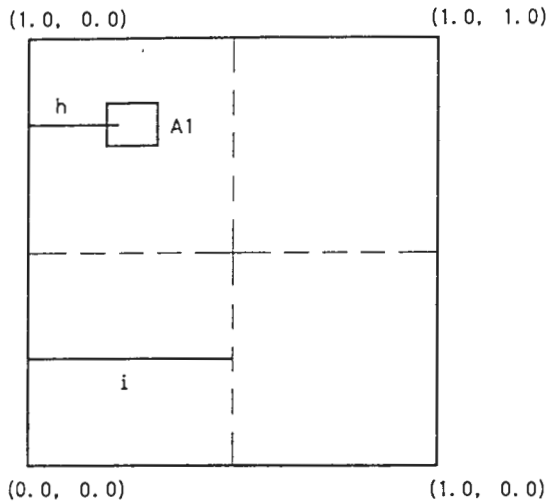
$$m = h / i$$

$$\text{val} = (1.0 - m) * \exp(-a * m)$$

a はチューニングの為のパラメータです.

4.5 左 (空間指示 sleft)

『左』の対象となる図形の条件は, 図形の右端が Frame の中心より左にある物のみ.



計算に必要な h, i の関係図

$\pi / 2 \leq \text{azimuth2}(\text{Frame A1}) \leq \pi$ または
 $-\pi \leq \text{azimuth2}(\text{Frame A1}) \leq -\pi / 2$ の時

$h = x1$
 $i = 0.5$

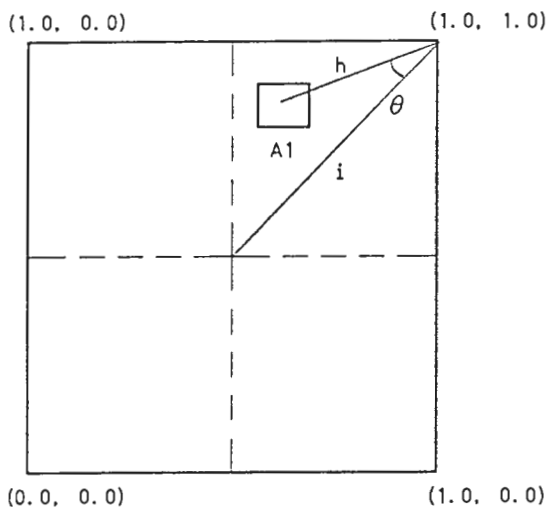
$m = h / i$

$\text{val} = (1.0 - m) * \exp(-a * m)$

a はチューニングの為のパラメータです.

4.6 右上 (空間指示 stop-sright)

『右上』の対象となる図形の条件は, 図形の左端が Frame の中心より右にあり, かつ図形の下端が Frame の中心より上にある物のみ.



計算に必要な h, i の関係図

$-\pi / 2 \leq \text{azimuth3}(\text{Frame A1}) \leq \pi / 2$ かつ
 $0 \leq \text{azimuth5}(\text{Frame A1}) \leq \pi$ の時

$h = \sqrt{(1.0 - x1) * (1.0 - x1) + (1.0 - y1) * (1.0 - y1)}$

$i = \sqrt{0.5 * 0.5 + 0.5 * 0.5}$

$m = i / h$

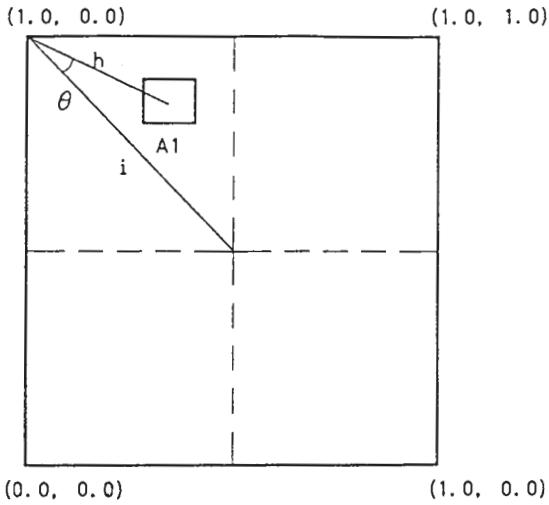
$n = (1.0 - m) * \exp(-a * m)$

$\text{val} = n * (1.0 - (|\theta + 3\pi / 4|) / (3\pi / 4))$

a はチューニングの為のパラメータです.

4.7 左上 (空間指示 stop-sleft)

『左上』の対象となる図形の条件は, 図形の右端が Frame の中心より左にあり, かつ図形の下端が Frame の中心より上にある物のみ.



計算に必要な h, i の関係図

- 1) $\pi / 2 \leq \text{azimuth2}(\text{Frame A1}) \leq \pi$ または $-\pi \leq \text{azimuth2}(\text{Frame A1}) \leq -\pi / 2$
- 2) $0 \leq \text{azimuth5}(\text{Frame A1}) \leq \pi$

1) かつ 2) が成り立つ時

$$h = \sqrt{x1*x1 + (1.0 - y1)*(1.0 - y1)}$$

$$i = \sqrt{0.5 * 0.5 + 0.5 * 0.5}$$

$$m = i / h$$

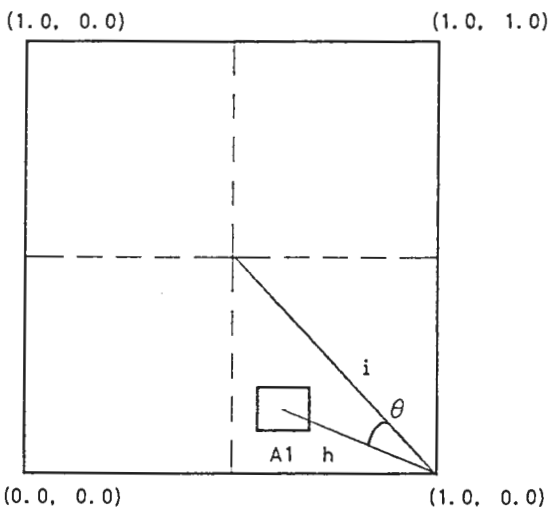
$$n = (1.0 - m) * \exp(-a * m)$$

$$\text{val} = n*(1.0 - (|\theta + \pi / 4|) / (\pi / 4))$$

a はチューニングの為のパラメータです.

4.8 右下 (空間指示 sbottom-sright)

『右下』の対象となる図形の条件は, 図形の左端が Frame の中心より右にあり, かつ図形の上端が Frame の中心より下にある物のみ.



計算に必要な h, i の関係図

- 1) $-\pi \leq \text{azimuth4}(\text{Frame A1}) \leq 0$
- 2) $-\pi / 2 \leq \text{azimuth5}(\text{Frame A1}) \leq \pi / 2$

1) かつ 2) が成り立つ時

$$h = \sqrt{(1.0 - x1)*(1.0 - x1) + y1*y1}$$

$$i = \sqrt{0.5 * 0.5 + 0.5 * 0.5}$$

$$m = i / h$$

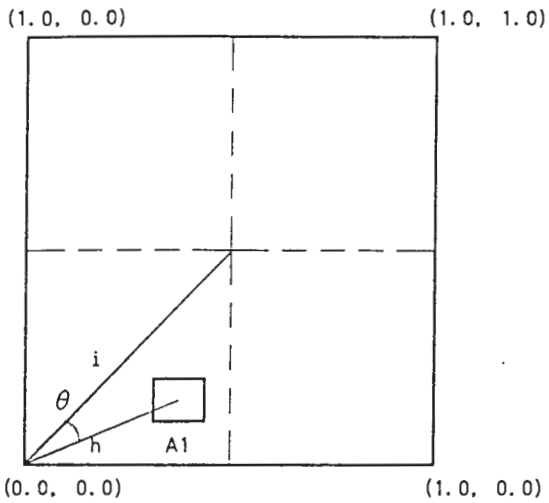
$$n = (1.0 - m) * \exp(-a * m)$$

$$\text{val} = n*(1.0 - (|\theta - 3 \pi / 4|) / (3 \pi / 4))$$

a はチューニングの為のパラメータです.

4.9 左下 (空間指示 sbottom-left)

『左下』の対象となる図形の条件は, 図形の右端が Frame の中心より左にあり, かつ図形の上端が Frame の中心より下にある物のみ.



計算に必要な h, i の関係図

- 1) $\pi / 2 \leq \text{azimuth2}(\text{Frame A1}) \leq \pi$ または
 $-\pi \leq \text{azimuth2}(\text{Frame A1}) \leq -\pi / 2$
- 2) $-\pi \leq \text{azimuth4}(\text{Frame A1}) \leq 0$

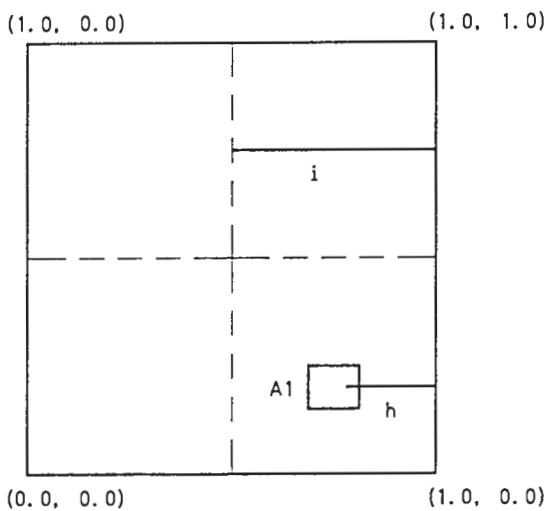
1) かつ 2) が成り立つ時

```

h = sqrt(x1 * x1 + y1 * y1)
i = sqrt(0.5 * 0.5 - 0.5 * 0.5)
m = i / h
n = (1.0 - m) * exp(-a * m)
val = n * (1.0 - (|theta - pi / 4|) / (pi / 4))
a はチューニングの為のパラメータです.
    
```

4.10 横端 (空間指示 edge-x)

『横端』の対象となる図形の条件チェックはない.



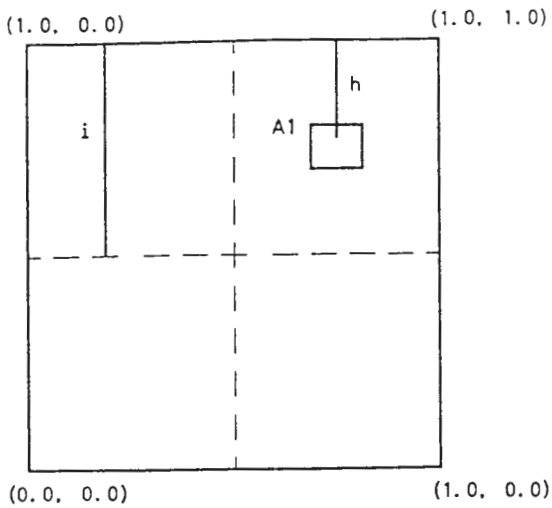
計算に必要な h, i の関係図

```

h = 1.0 - x1
i = 0.5
m = h / i
val = (1.0 - m) * exp(-a * m)
a はチューニングの為のパラメータです.
    
```

4.11 縦端 (空間指示 edge-y)

『縦端』の対象となる図形の条件チェックはない。



$$h = 1.0 - y_1$$

$$i = 0.5$$

$$m = h / i$$

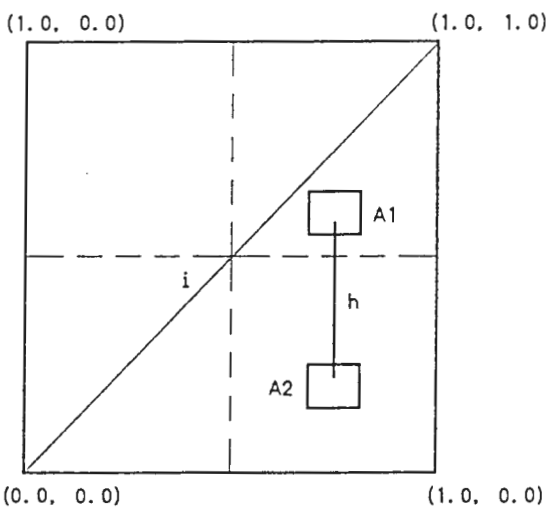
$$val = (1.0 - m) * \exp(-a * m)$$

a はチューニングの為のパラメータです。

計算に必要な h, i の関係図

4.12 近く (相対指示 near) / 遠く (相対指示 far)

『近く』『遠く』の対象となる図形の条件チェックはない。



$$h = \text{distance2}(A1, A2)$$

$$i = \text{sqrt}(1.0 * 1.0 + 1.0 * 1.0)$$

近くの場合

$$val = 1.0 - h / i$$

遠くの場合

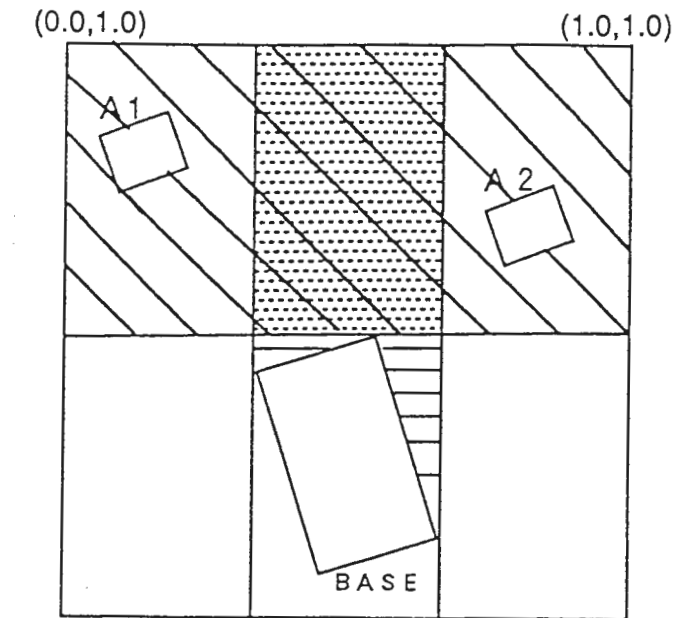
$$val = h / i$$

計算に必要な h, i の関係図

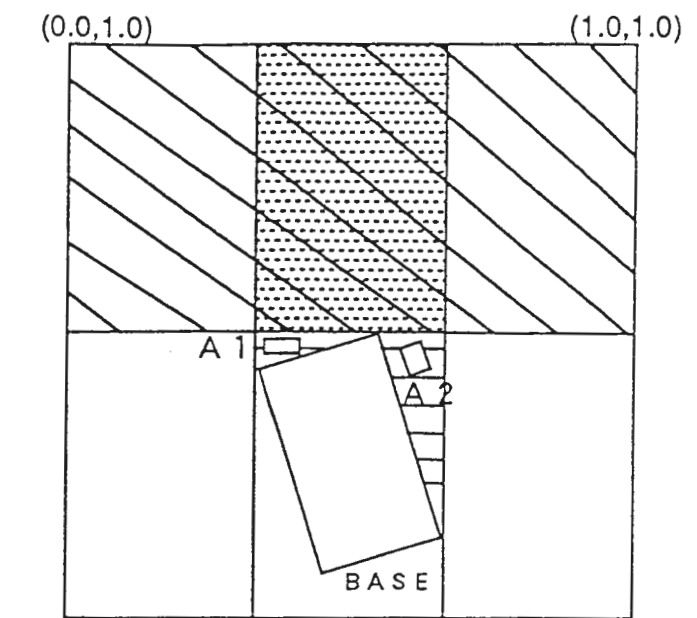
4.13 上 (相対指示 up)

1. 基準図形からみて『上』の対象となる図形の条件
基準図形に対して LEVEL2 の remote もしくは touching の関係を満たし、以下の3つのいずれかの条件を満たす図形が『上』の対象となる。
 - (a) 図の領域 A に図形が含まれる。
 - (b) 図の領域 B に図形が含まれる。この条件のチェックは shi-cf-calc.lisp の test-position という関数を Call して行なっている。
 - (c) 図の領域 C に図形の一部が含まれる。この条件のチェックは shi-cf-calc.lisp の test-cross という関数を Call して行なっている。
2. 基準図形に対する『上』の対象となる図形の『上らしさ』は以下のように数値化する。
 - (a) 『上らしさ』の計算は likeup.4.02.c(C program) で行なわれる。
 - (b) 対象図形の位置データは work2 ファイルの CENTER-RULE, PBOUND-RULE から得る。
 - (c) 対象図形 (複数) から任意に 2 図形選び、基準図形と 2 図形間の角度と距離データより、基準図形に対して 2 図形のどちらがどれだけ『上らしい』かを以下のように求める。
 - 2 図形間の『上らしさ』の計算方法
 - i. 基準図形の中心点と図形 1 の基準図形に対する最も『上らしい』点との距離、角度を r_1, θ_1 とする¹。
 - ii. 基準図形の中心点と図形 2 の基準図形に対する最も『上らしい』点との距離、角度を r_2, θ_2 とする。
 - iii. 但し、基準図形に対して $|\text{角度} - \pi/2|$ が大きい方 (角度は $-\pi$ から π の区間に存在する) を図形 2 とする。
 - iv. $R_1 = r_1 / r_2$
 - v. $\theta = (|\theta_2 - \pi/2| - |\theta_1 - \pi/2|) / (\pi/2)$
 - vi. $R_2 = (10 / \exp(b * \theta)) + 1$
 - vii. $R_3 = R_1 - ((1 - \theta) / \exp(a * \theta))$
 - viii. $Z = R_2 * R_3 * R_3 * R_3$
 - a, b はチューニングの為のパラメータです。
 - (d) これをすべての組合せ回数繰り返す。
(例) 図形が base, A1, A2, A3 の時
(A1, A2), (A1, A3), (A2, A3) の組合せ数回 (3回) 繰り返す。
 - (e) 求められた値を $-0.5 \sim 0.5$ の間に正規化する。
 - (f) 正規化された値を正規分布表により変換する。
 - (g) 求められた各々の値を、一対比較法により順位付けを行なう。
 - (h) 順位付けされた値を正規化しそれを関数値 (val) とする。

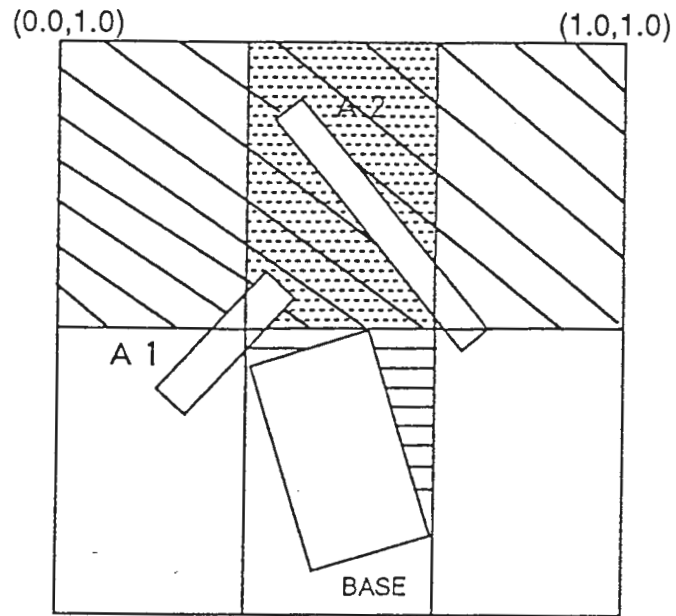
¹最も『上らしい』計算方法は付録 A を参照のこと。



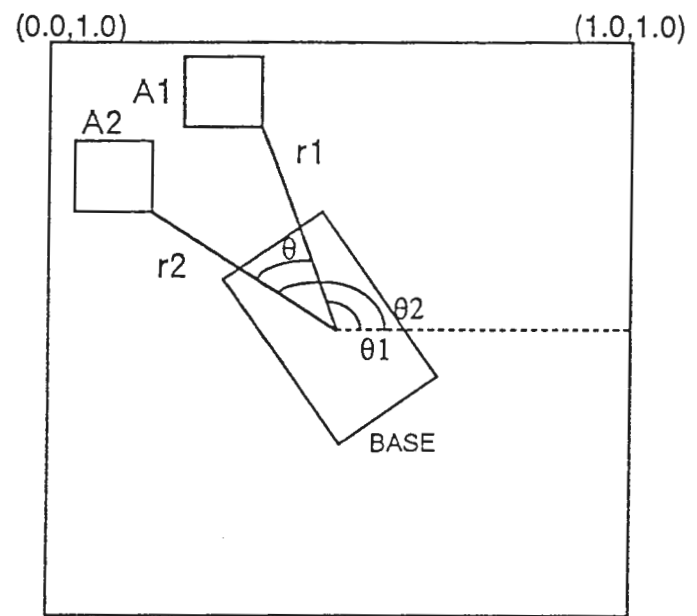
(0.0,0.0) (1.0,0.0)
領域Aに図形が含まれる場合



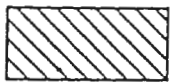
(0.0,0.0) (1.0,0.0)
領域Bに図形が含まれる場合



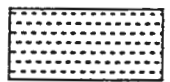
(0.0,0.0) (1.0,0.0)
領域Cに図形が含まれる場合



(0.0,0.0) (1.0,0.0)
図形の「上らしさ」計算のための
距離、角度の関係



領域A



領域B

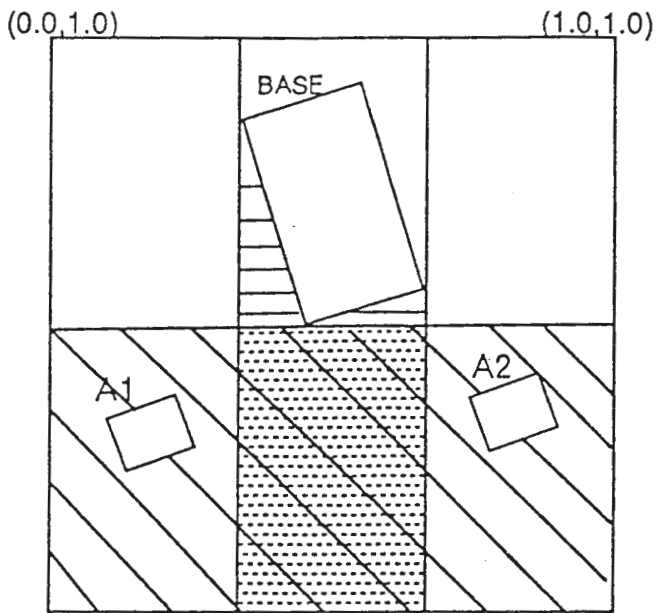


領域C

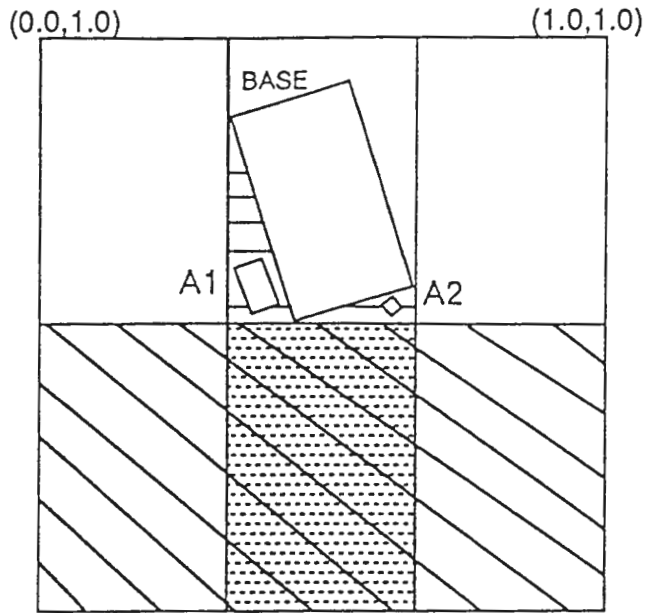
4.14 下 (相対指示 down)

1. 基準図形からみて『下』の対象となる図形の条件
基準図形に対して LEVEL2 の remote もしくは touching の関係を満たし、以下の3つのいずれかの条件を満たす図形が『下』の対象となる。
 - (a) 図の領域 A に図形が含まれる。
 - (b) 図の領域 B に図形が含まれる。この条件のチェックは shi-cf-calc.lisp の test-position という関数を Call して行なっている。
 - (c) 図の領域 C に図形の一部が含まれる。この条件のチェックは shi-cf-calc.lisp の test-cross という関数を Call して行なっている。
2. 基準図形に対する『下』の対象となる図形の『下らしさ』は以下のように数値化する。
 - (a) 『下らしさ』の計算は likedown.4.02.c(C program) で行なわれる。
 - (b) 対象図形の位置データは work2 ファイルの CENTER-RULE, PBOUND-RULE から得る。
 - (c) 対象図形 (複数) から任意に 2 図形選び、基準図形と 2 図形間の角度と距離データより、基準図形に対して 2 図形のどちらがどれだけ『下らしい』かを以下のように求める。
 - 2 図形間の『下らしさ』の計算方法
 - i. 基準図形の中心点と図形 1 の基準図形に対する最も『下らしい』点との距離、角度を r_1, θ_1 とする²。
 - ii. 基準図形の中心点と図形 2 の基準図形に対する最も『下らしい』点との距離、角度を r_2, θ_2 とする。
 - iii. 但し、基準図形に対して $|\theta_1 - \pi/2|$ が大きい方 (角度は $-\pi$ から π の区間に存在する) を図形 2 とする。
 - iv. $R_1 = r_1 / r_2$
 - v. $\theta = (|\theta_2| - \pi/2 - |\theta_1| - \pi/2) / (\pi/2)$
 - vi. $R_2 = (10 / \exp(b * \theta)) + 1$
 - vii. $R_3 = R_1 - ((1 - \theta) / \exp(a * \theta))$
 - viii. $Z = R_2 * R_3 * R_3 * R_3$
 - a, b はチューニングの為のパラメータです。
 - (d) これをすべての組合せ回数繰り返す。
(例) 図形が base, A1, A2, A3 の時
(A1, A2), (A1, A3), (A2, A3) の組合せ数回 (3回) 繰り返す。
 - (e) 求められた値を $-0.5 \sim 0.5$ の間に正規化する。
 - (f) 正規化された値を正規分布表により変換する。
 - (g) 求められた各々の値を、一対比較法により順位付けを行なう。
 - (h) 順位付けされた値を正規化しそれを関数値 (val) とする。

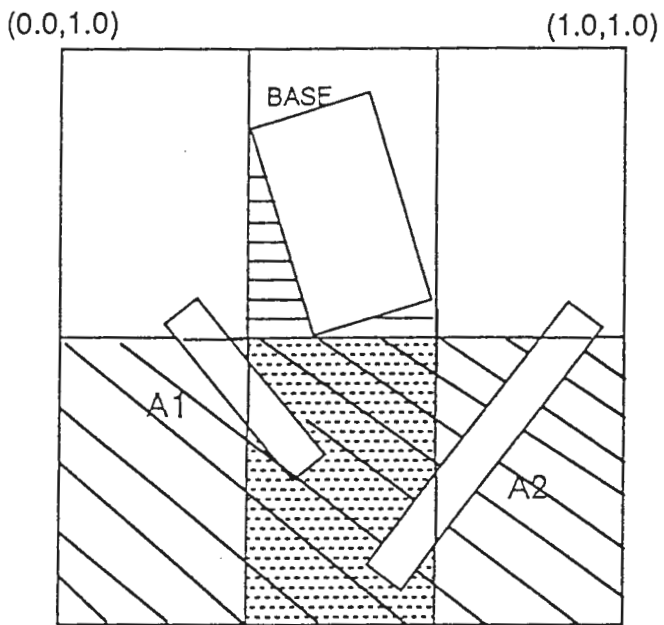
²最も『下らしい』計算方法は付録 B を参照のこと。



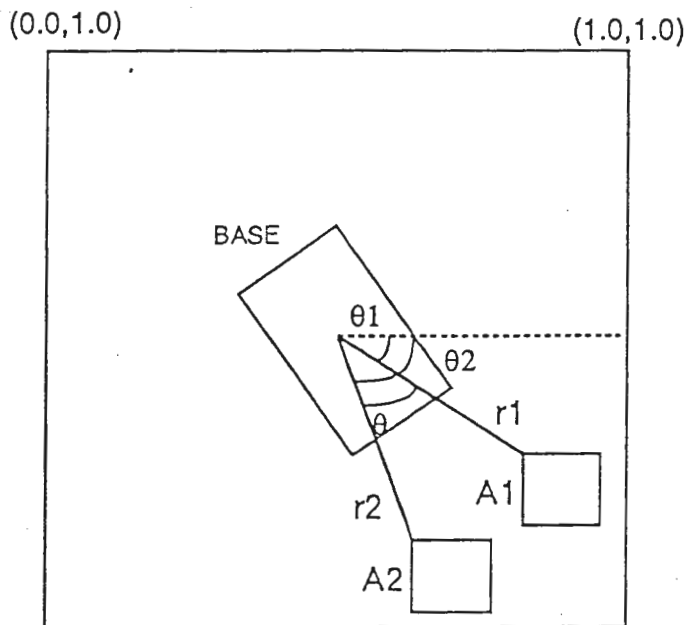
(0,0,0.0) (1,0,0.0)
領域Aに図形が含まれる場合



(0,0,0.0) (1,0,0.0)
領域Bに図形が含まれる場合



(0,0,0.0) (1,0,0.0)
領域Cに図形が含まれる場合



(0,0,0.0) (1,0,0.0)
図形の「下らしさ」計算のための
距離、角度の関係



領域A



領域B



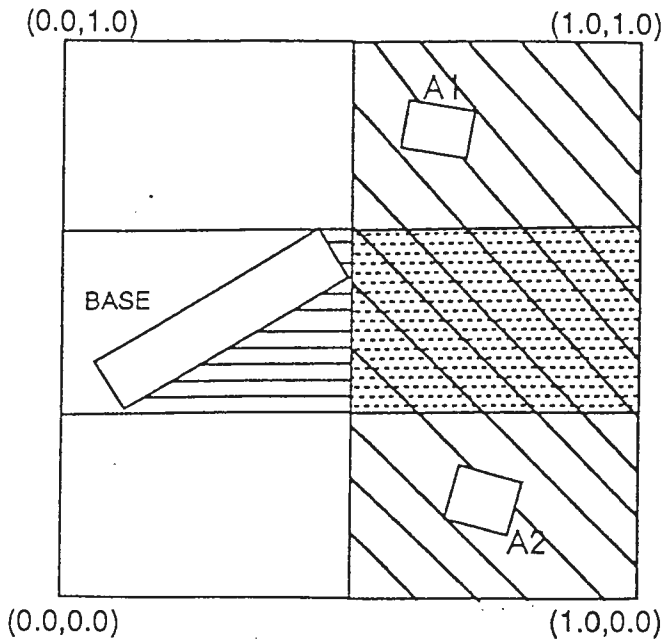
領域C

4.15 右 (相対指示 right)

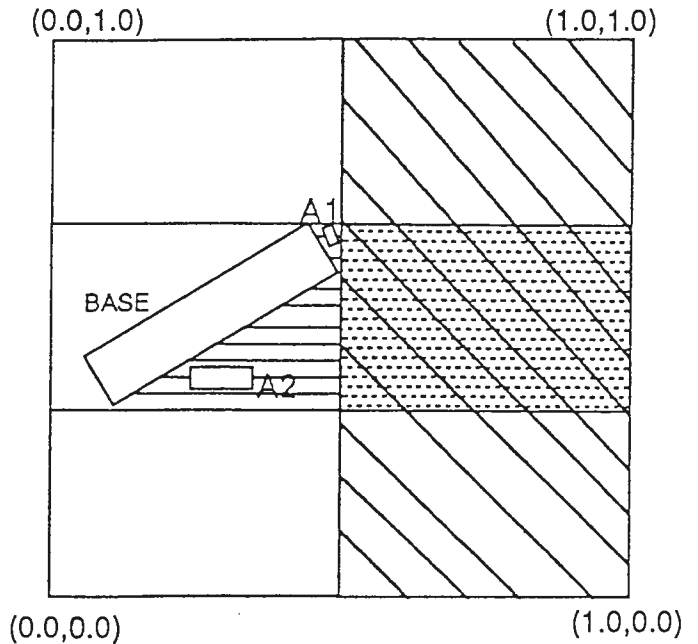
1. 基準図形からみて『右』の対象となる図形の条件
基準図形に対して LEVEL2 の remote もしくは touching の関係を満たし、以下の3つのいずれかの条件を満たす図形が『右』の対象となる。
 - (a) 図の領域 A に図形が含まれる.
 - (b) 図の領域 B に図形が含まれる. この条件のチェックは shi-cf-calc.lisp の test-position という関数を Call して行なっている.
 - (c) 図の領域 C に図形の一部が含まれる. この条件のチェックは shi-cf-calc.lisp の test-cross という関数を Call して行なっている.
2. 基準図形に対する『右』の対象となる図形の『右らしさ』は以下のように数値化する.
 - (a) 『右らしさ』の計算は likeright.5.02.c(C program) で行なわれる.
 - (b) 対象図形の位置データは work2 ファイルの CENTER-RULE, PBOUND-RULE から得る.
 - (c) 対象図形 (複数) から任意に 2 図形選び、基準図形と 2 図形間の角度と距離データより、基準図形に対して 2 図形のどちらがどれだけ『右らしい』かを以下のように求める.
 - 2 図形間の『右らしさ』の計算方法
 - i. 基準図形の中心点と図形 1 の基準図形に対する最も『右らしい』点との距離、角度を r_1, θ_1 とする³.
 - ii. 基準図形の中心点と図形 2 の基準図形に対する最も『右らしい』点との距離、角度を r_2, θ_2 とする.
 - iii. 但し、基準図形に対して | 角度 | が大きい方 (角度は $-\pi$ から π の区間に存在する) を図形 2 とする.
 - iv. $R_1 = r_1 / r_2$
 - v. $\theta = (|\theta_2| - |\theta_1|) / (\pi / 2)$
 - vi. $R_2 = (10 / \exp(b * \theta)) + 1$
 - vii. $R_3 = R_1 - ((1 - \theta) / \exp(a * \theta))$
 - viii. $Z = R_2 * R_3 * R_3 * R_3$
 - a, b はチューニングの為のパラメータです.
 - (d) これをすべての組合せ回数繰り返す.
(例) 図形が base, A1, A2, A3 の時
(A1, A2), (A1, A3), (A2, A3) の組合せ数回 (3回) 繰り返す.
 - (e) 求められた値を $-0.5 \sim 0.5$ の間に正規化する.
 - (f) 正規化された値を正規分布表により変換する.
 - (g) 求められた各々の値を、一対比較法により順位付けを行なう.
 - (h) 順位付けされた値を正規化しそれを関数値 (val) とする.

³最も『右らしい』計算方法は付録 C を参照のこと.

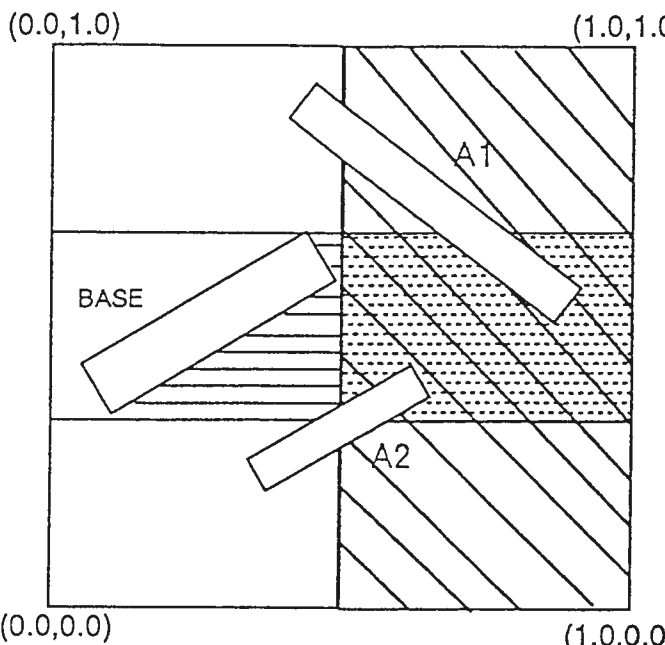
right



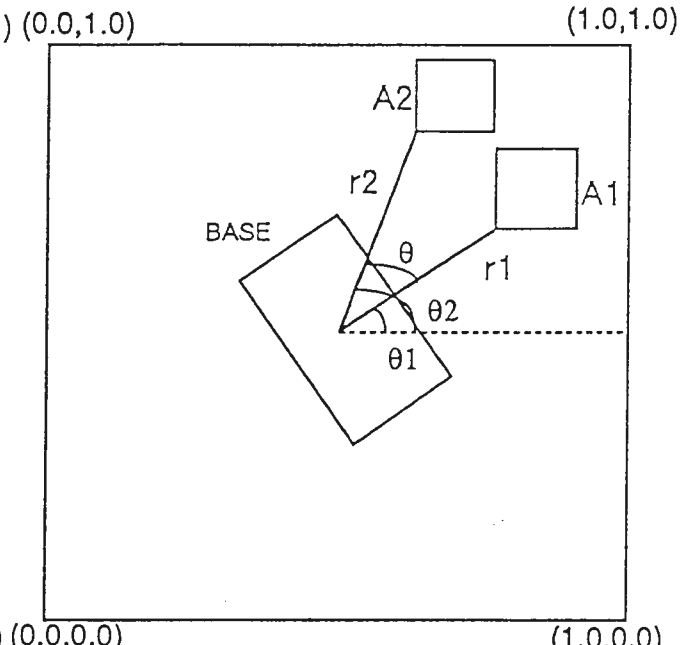
領域Aに図形が含まれる場合



領域Bに図形が含まれる場合



領域Cに図形が含まれる場合



図形の『右らしさ』計算のための距離、角度の関係



領域A



領域B

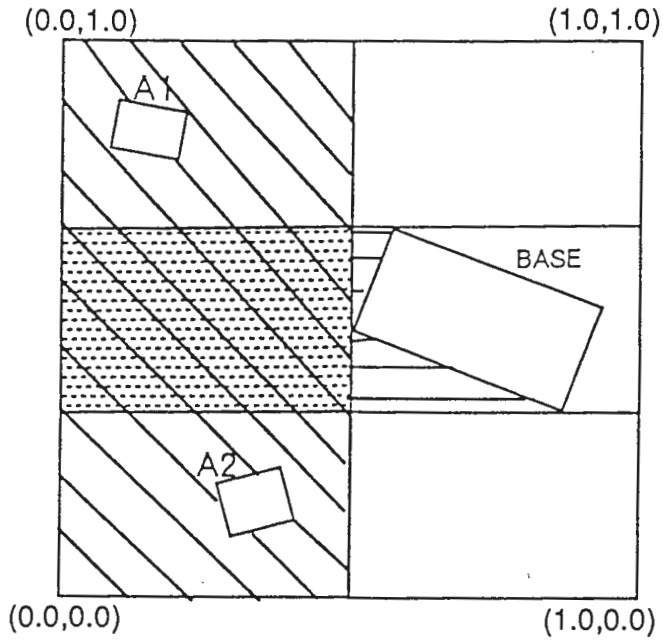


領域C

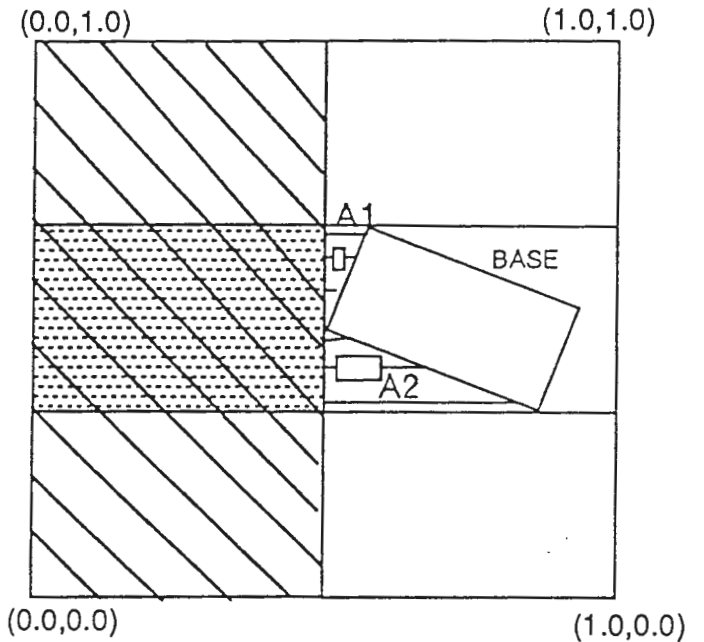
4.16 左 (相対指示 left)

1. 基準図形からみて『左』の対象となる図形の条件
基準図形に対して LEVEL2 の remote もしくは touching の関係を満たし、以下の3つのいずれかの条件を満たす図形が『左』の対象となる。
 - (a) 図の領域 A に図形が含まれる。
 - (b) 図の領域 B に図形が含まれる。この条件のチェックは shi-cf-calc.lisp の test-position という関数を Call して行なっている。
 - (c) 図の領域 C に図形の一部が含まれる。この条件のチェックは shi-cf-calc.lisp の test-cross という関数を Call して行なっている。
2. 基準図形に対する『左』の対象となる図形の『左らしさ』は以下のように数値化する。
 - (a) 『左らしさ』の計算は likeleft.4.02.c(C program) で行なわれる。
 - (b) 対象図形の位置データは work2 ファイルの CENTER-RULE, PBOUND-RULE から得る。
 - (c) 対象図形 (複数) から任意に 2 図形選び、基準図形と 2 図形間の角度と距離データより、基準図形に対して 2 図形のどちらがどれだけ『左らしい』かを以下のように求める。
 - 2 図形間の『左らしさ』の計算方法
 - i. 基準図形の中心点と図形 1 の基準図形に対する最も『左らしい』点との距離、角度を r_1, θ_1 とする⁴。
 - ii. 基準図形の中心点と図形 2 の基準図形に対する最も『左らしい』点との距離、角度を r_2, θ_2 とする。
 - iii. 但し、基準図形に対して $|\theta_1 - \theta_2|$ が大きい方 (角度は $-\pi$ から π の区間に存在する) を図形 2 とする。
 - iv. $R_1 = r_1 / r_2$
 - v. $\theta = (|\theta_2| - \pi - |\theta_1| - \pi) / (\pi / 2)$
 - vi. $R_2 = (10 / \exp(b * \theta)) + 1$
 - vii. $R_3 = R_1 - ((1 - \theta) / \exp(a * \theta))$
 - viii. $Z = R_2 * R_3 * R_3 * R_3$
 - a, b はチューニングの為のパラメータです。
 - (d) これをすべての組合せ回数繰り返す。
(例) 図形が base, A1, A2, A3 の時
(A1, A2), (A1, A3), (A2, A3) の組合せ数回 (3回) 繰り返す。
 - (e) 求められた値を -0.5 ~ 0.5 の間に正規化する。
 - (f) 正規化された値を正規分布表により変換する。
 - (g) 求められた各々の値を、一対比較法により順位付けを行なう。
 - (h) 順位付けされた値を正規化しそれを関数値 (val) とする。

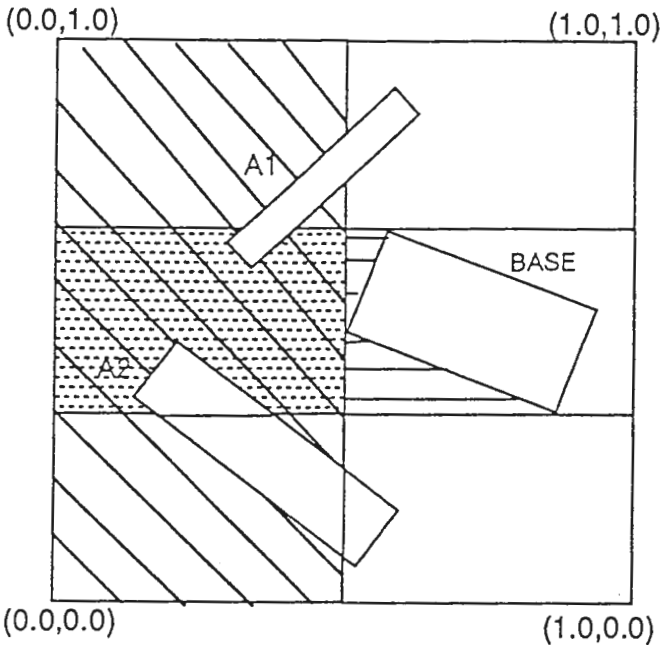
⁴最も『左らしい』計算方法は付録 C を参照のこと。



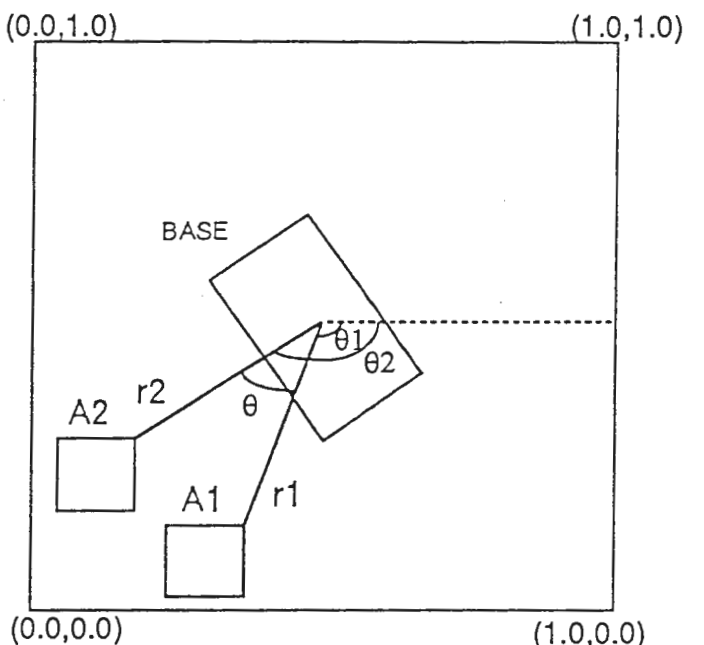
領域Aに図形が含まれる場合






領域Bに図形が含まれる場合



領域Cに図形が含まれる場合



図形の『左らしさ』計算のための距離、角度の関係

-  領域A
-  領域B
-  領域C

4.17 重なる (相対指示 overlap)

within または crossing であれば、val 値は 1.0 である。

4.18 隣 (相対指示 neighbor)

1. 基準となる図形から最も近い図形の距離を求める。
2. その距離を 100 とした時に 110 の距離までの図形を検索する。
3. 検索された各々の図形の val 値の合計が 1.0 になるように 80% は等分に分配し、残りの 20% は距離の比で分配する。

(例) 図形は base, A1, A2, A3, A4 の 5 種類とし、base からの隣の度合を計算する。

```
L1 = 0.100 = distance2(base, A1)
L2 = 0.105 = distance2(base, A2)
L3 = 0.110 = distance2(base, A3)
L4 = 0.115 = distance2(base, A4)
```

とした時、最短距離は L1 の 0.1 である。

```
L1 : L2 = 0.100 : 0.105 = 100 : 105
L1 : L3 = 0.100 : 0.110 = 100 : 110
L1 : L4 = 0.100 : 0.115 = 100 : 115
```

より L4 は対象外となる。

まず val 値を L1, L2, L3 各々に $0.8/3 = 0.266666$ ずつ分配する。

```
X = ((L1 * 1.1 - L1) + (L1 * 1.1 - L2) + (L1 * 1.1 - L3)) / L1
```

```
val(A1) = 0.4000 = 0.266666 + (0.2 * (L1 * 1.1 - L1) / X)
val(A2) = 0.3333 = 0.266666 + (0.2 * (L1 * 1.1 - L2) / X)
val(A3) = 0.2666 = 0.266666 + (0.2 * (L1 * 1.1 - L3) / X)
```

ゆえに base に対する隣の度合は

```
A1 : 0.40
A2 : 0.33
A3 : 0.27
```

となる。

val の計算は shi-cf-calc の val-neighbor より neighbor(C program) を Call して行なっている。

4.19 ダミー (dummy2)

1. このルールは強制的に CENTER-RULE, PBOUND-RULE, DISTANCE2-RULE を起動して
(rel 0 center ...) (rel 0 pbound ...) (rel 0 distance2 ...)
の関係を work2 ファイルに出力する為に作成する.
2. このルールの定義は level2.lisp ファイルの一番最後で定義するものとする.SPADE システムでは最後に定義したルールが最初に起動されるため, 最初に work2 ファイルに必要なデータを格納することができる.

5 LEVEL3 のルール

- level3.lisp 内で定義している.
- LEVEL3 の RULE の FACT 表現は以下の通りです.
(rel 3 rel-name (objects) (nil) cf 値)
- LEVEL3 では LEVEL2 の val 値を元に cf 値を計算している.

以下に LEVEL3 での cf 値の計算方法を示す.

5.1 一般的なルール

以下に示すものは、LEVEL2 のルールの val 値をそのまま LEVEL3 の cf 値とする.

LEVEL3 のルール名	LEVEL2 のルールの名
1. 上 (空間指示 S 上) ……………	stop
2. 下 (空間指示 S 下) ……………	sbottom
3. 右 (空間指示 S 右) ……………	sright
4. 左 (空間指示 S 左) ……………	sleft
5. 中央 (空間指示 S 中央) ……	smiddle
6. 右下 (空間指示 S 右下) ……	sbottom-sright
7. 右上 (空間指示 S 右上) ……	stop-sright
8. 左下 (空間指示 S 左下) ……	sbottom-sleft
9. 左上 (空間指示 S 左上) ……	stop-sleft
10. 右端 (空間指示 S 右端) ……	sright
11. 左端 (空間指示 S 左端) ……	sleft
12. 上端 (空間指示 S 上端) ……	stop
13. 下端 (空間指示 S 下端) ……	sbottom
14. 上 (相対指示 上) ……………	up
15. 下 (相対指示 下) ……………	down
16. 右 (相対指示 右) ……………	right
17. 左 (相対指示 左) ……………	left
18. 重 (相対指示 重) ……………	overlap
19. 隣 (相対指示 隣) ……………	neighbor

5.2 端 (空間指示 S 端)

LEVEL2 の edge-x と edge-y との val 値の大きい方を cf 値とする.

5.3 隅 (空間指示 S 隅)

LEVEL2 の edge-x と edge-y との val 値を掛けた物を cf 値とする.

5.4 横 (相対指示 横)

LEVEL2 の sright か sleft の val 値の大きい方を cf 値とする.

5.5 ダミー

LEVEL2 の dummy2 を起動させる為のルールである. (LEVEL2 の dummy2 参照の事).

6 shi-cf-calc.lisp について

本節では shi-cf-calc.lisp ファイルにおいて定義している cf 値を求める関数についての説明を行なう。

6.1 cf-stop

空間指示の上 (STOP) の val 値を求める。

6.2 cf-sbottom

空間指示の下 (SBOTTOM) の val 値を求める。

6.3 cf-sright

空間指示の右 (SRIGHT) の val 値を求める。

6.4 cf-sleft

空間指示の左 (SLEFT) の val 値を求める。

6.5 cf-smiddle

空間指示の中央 (SMIDDLE) の val 値を求める。

6.6 cf-stop-sright

空間指示の右上 (STOP-SRIGHT) の val 値を求める。

6.7 cf-stop-sleft

空間指示の左上 (STOP-SLEFT) の val 値を求める。

6.8 cf-sbottom-sright

空間指示の右下 (SBOTTOM-SRIGHT) の val 値を求める。

6.9 cf-sbottom-sleft

空間指示の左下 (SBOTTOM-SLEFT) の val 値を求める。

6.10 val-edge

LEVEL2 の X-EDGE, Y-EDGE の val 値を求める。

6.11 hashi-calc

空間指示の端の val 値を求める。
X-EDGE, Y-EDGE の val 値の大きい方を val 値とする。

6.12 sumi-calc

空間指示の隅の val 値を求める。
X-EDGE, Y-EDGE の val 値の積を val 値とする。

6.13 cf-up

相対指示の上の val 値を求める.

likeup(C PROGRAM) を Call し val 値を計算している.

6.14 cf-down

相対指示の下の val 値を求める.

likedown(C PROGRAM) を Call し val 値を計算している.

6.15 cf-right

相対指示の右の val 値を求める.

likeright(C PROGRAM) を Call し val 値を計算している.

6.16 cf-left

相対指示の左の val 値を求める.

likeleft(C PROGRAM) を Call し val 値を計算している.

6.17 likesup

likeup を Call する関数.

6.18 likesdown

likedown を Call する関数.

6.19 likesright

likeright を Call する関数.

6.20 likesleft

likeleft を Call する関数.

6.21 val-neighbor

相対指示の隣 (NEIGHBOR) の val 値を計算する.

neighbor(C PROGRAM) を Call し val 値を計算している.

6.22 test-cross

対指示の上下左右の成立条件の判断を行なっている.

基準となる図形の TOP-BOUND, BOTTOM-BOUND からの水平延長線と RIGHT-BOUND, LEFT-BOUND からの垂直延長線とに対して対象図形の各辺が交差しているかどうかのチェックをしている.

6.23 test-position

相対指示の上下左右の成立条件の判断を行なっている.

対象図形が基準となる図形の上下左右どちら側にあるかのチェックをしている.

A 付録『上らしさ』の計算方法

基準図形に対して、ある図形の端点および中心点のうちどれが最も『上らしい』かは、以下の様にして計算する。

1. 基準図形からみて『上』の対象となる点の条件は、基準図形の中心点より『上』にある点とする。
2. 上記の条件を満たす点の基準図形に対する『上らしさ』は以下のように数値化する。

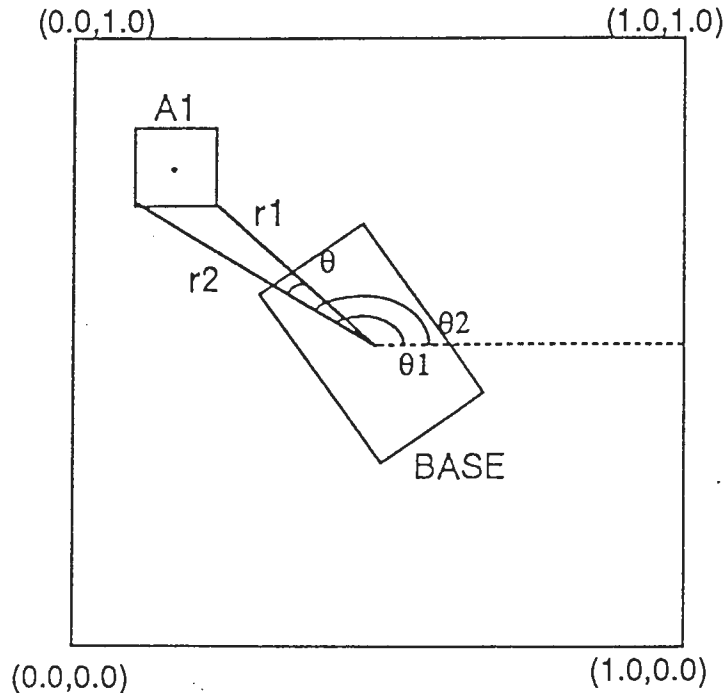
- (a) 対象となる点(複数)から任意に2点選び、基準図形と2点間の角度と距離データより、基準図形に対して2点のどちらがどれだけ『上らしい』かを以下のように求める。

2点間の『上らしさ』の計算方法

- i. 基準図形の中心点と点1との距離、角度を r_1, θ_1 とする。
- ii. 基準図形の中心点と点2との距離、角度を r_2, θ_2 とする。
- iii. 但し、基準図形に対して $|\text{角度} - \pi/2|$ が大きい方(角度は $-\pi$ から π の区間に存在する)を点2とする。
- iv. $R_1 = r_1 / r_2$
- v. $\theta = (|\theta_2 - \pi/2| - |\theta_1 - \pi/2|) / (\pi/2)$
- vi. $R_2 = (10 / \exp(b * \theta)) + 1$
- vii. $R_3 = R_1 - (1 - \theta) / \exp(a * \theta)$
- viii. $Z = R_2 * R_3 * R_3 * R_3$

a, b はチューニングの為のパラメータです。

- (b) これをすべての組合せ回数繰り返す。
 - (c) 求められた値を $-0.5 \sim 0.5$ の間に正規化する。
 - (d) 正規化された値を正規分布表により変換する。
 - (e) 求められた各々の値を、一対比較法により順位付けを行なう。
3. 順位付けされた値の一番大きい点を目的とする点とする。



『上らしい』点の計算ための距離、
角度の関係

B 付録『下らしさ』の計算方法

基準図形に対して、ある図形の端点および中心点のうちどれが最も『下らしい』かは、以下の様にして計算する。

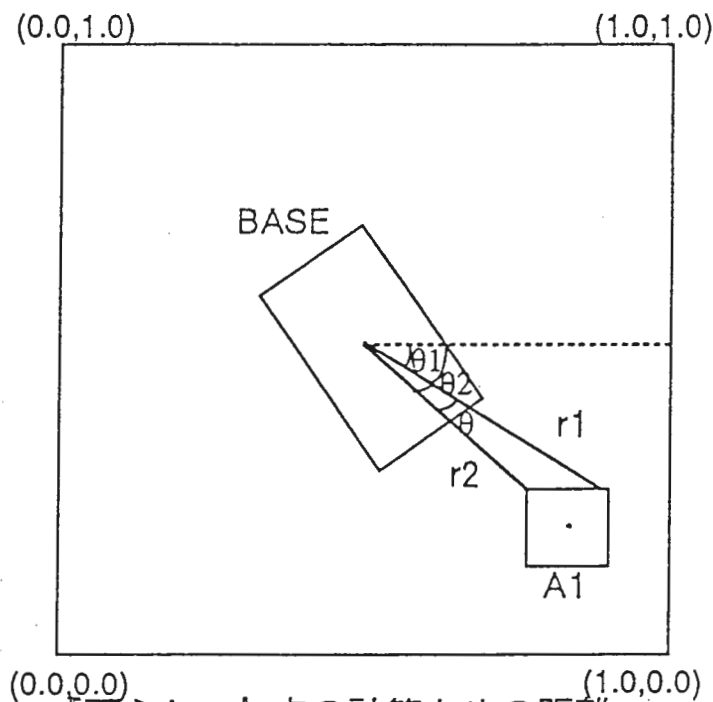
1. 基準図形からみて『下』の対象となる点の条件は、基準図形の中心点より『下』にある点とする。
2. 上記の条件を満たす点の基準図形に対する『下らしさ』は以下のように数値化する。

(a) 対象となる点(複数)から任意に2点選び、基準図形と2点間の角度と距離データより、基準図形に対して2点のどちらがどれだけ『下らしい』かを以下のように求める。

2点間の『下らしさ』の計算方法

- i. 基準図形の中心点と点1との距離、角度を r_1, θ_1 とする。
 - ii. 基準図形の中心点と点2との距離、角度を r_2, θ_2 とする。
 - iii. 但し、基準図形に対して $|\theta - \pi/2|$ が大きい方(角度は $-\pi$ から π の区間に存在する)を点2とする。
 - iv. $R_1 = r_1 / r_2$
 - v. $\theta = (|\theta_2| - \pi/2 - (|\theta_1| - \pi/2)) / (\pi/2)$
 - vi. $R_2 = (10 / \exp(b * \theta)) + 1$
 - vii. $R_3 = R_1 - ((1 - \theta) / \exp(a * \theta))$
 - viii. $Z = R_2 * R_3 * R_3 * R_3$
- a, b はチューニングの為のパラメータです。

- (b) これをすべての組合せ回数繰り返す。
 - (c) 求められた値を $-0.5 \sim 0.5$ の間に正規化する。
 - (d) 正規化された値を正規分布表により変換する。
 - (e) 求められた各々の値を、一対比較法により順位付けを行なう。
3. 順位付けされた値の一番大きい点を目的とする点とする。



『下らしい』点の計算ための距離、
角度の関係

C 付録『右らしさ』の計算方法

基準図形に対して、ある図形の端点および中心点のうちどれが最も『右らしい』かは、以下の様にして計算する。

1. 基準図形からみて『右』の対象となる点の条件は、基準図形の中心点より『右』にある点とする。
2. 上記の条件を満たす点の基準図形に対する『右らしさ』は以下のように数値化する。

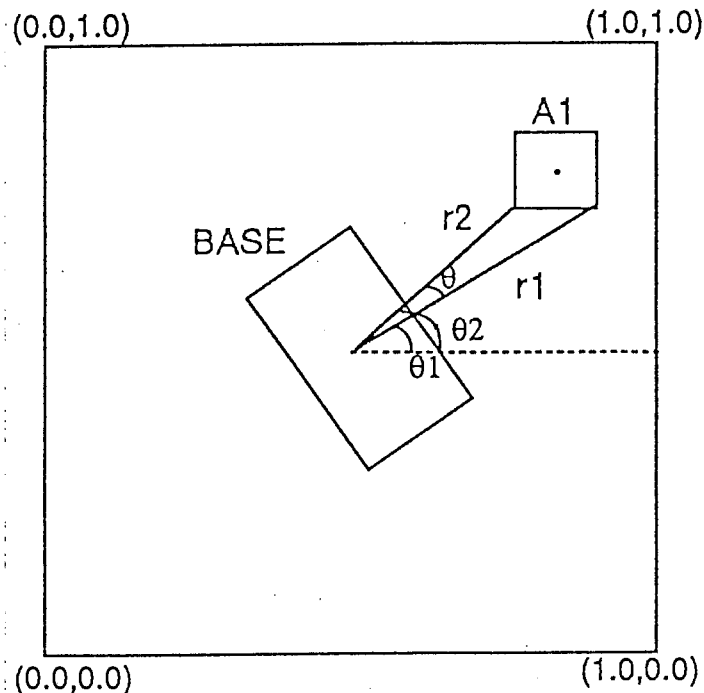
(a) 対象となる点(複数)から任意に2点選び、基準図形と2点間の角度と距離データより、基準図形に対して2点のどちらがどれだけ『右らしい』かを以下のように求める。

2点間の『右らしさ』の計算方法

- i. 基準図形の中心点と点1との距離、角度を r_1, θ_1 とする。
- ii. 基準図形の中心点と点2との距離、角度を r_2, θ_2 とする。
- iii. 但し、基準図形に対して | 角度 | が大きい方 (角度は $-\pi$ から π の区間に存在する) を点2とする。
- iv. $R_1 = r_1 / r_2$
- v. $\theta = (|\theta_2| - |\theta_1|) / (\pi / 2)$
- vi. $R_2 = (10 / \exp(b * \theta)) + 1$
- vii. $R_3 = R_1 - ((1 - \theta) / \exp(a * \theta))$
- viii. $Z = R_2 * R_3 * R_3 * R_3$

a, b はチューニングの為のパラメータです。

- (b) これをすべての組合せ回数繰り返す。
 - (c) 求められた値を $-0.5 \sim 0.5$ の間に正規化する。
 - (d) 正規化された値を正規分布表により変換する。
 - (e) 求められた各々の値に、一対比較法により順位付けを行なう。
3. 順位付けされた値の一番大きい点を目的とする点とする。



「右らしい」点の計算ための距離、角度の関係

D 付録『左らしさ』の計算方法

基準図形に対して、ある図形の端点および中心点のうちどれが最も『左らしい』かは、以下の様にして計算する。

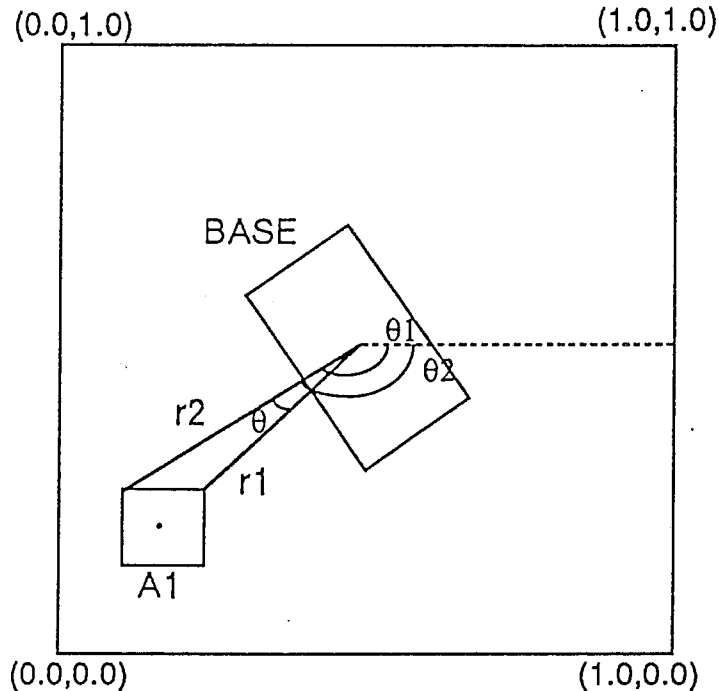
1. 基準図形からみて『左』の対象となる点の条件は、基準図形の中心点より『左』にある点とする。
2. 上記の条件を満たす点の基準図形に対する『左らしさ』は以下のように数値化する。

- (a) 対象となる点(複数)から任意に2点選び、基準図形と2点間の角度と距離データより、基準図形に対して2点のどちらがどれだけ『左らしい』かを以下のように求める。

2点間の『左らしさ』の計算方法

- i. 基準図形の中心点と点1との距離、角度を r_1, θ_1 とする。
 - ii. 基準図形の中心点と点2との距離、角度を r_2, θ_2 とする。
 - iii. 但し、基準図形に対して $|\theta - \pi|$ が大きい方(角度は $-\pi$ から π の区間に存在する)を点2とする。
 - iv. $R_1 = r_1 / r_2$
 - v. $\theta = (|\theta_2| - \pi - |\theta_1| - \pi) / (\pi / 2)$
 - vi. $R_2 = (10 / \exp(b * \theta)) + 1$
 - vii. $R_3 = R_1 - ((1 - \theta) / \exp(a * \theta))$
 - viii. $Z = R_2 * R_3 * R_3 * R_3$
- a, b はチューニングの為のパラメータです。

- (b) これをすべての組合せ回数繰り返す。
 - (c) 求められた値を $-0.5 \sim 0.5$ の間に正規化する。
 - (d) 正規化された値を正規分布表により変換する。
 - (e) 求められた各々の値に、一対比較法により順位付けを行なう。
3. 順位付けされた値の一番大きい点を目的とする点とする。



『左らしい』点の計算ための距離、角度の関係