

Object Classification – GetObject()

After Color and Entity Classification a raster image is converted in to a two dimensional list of Entities. The Entities in the image are loose ie not identified with an Object. Object Classification identifies Entities to Objects.

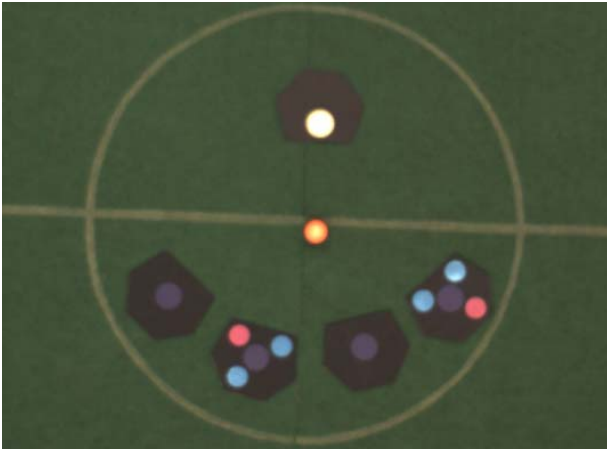


Figure 11.a: Raw image of soccer field w/ robots & ball.

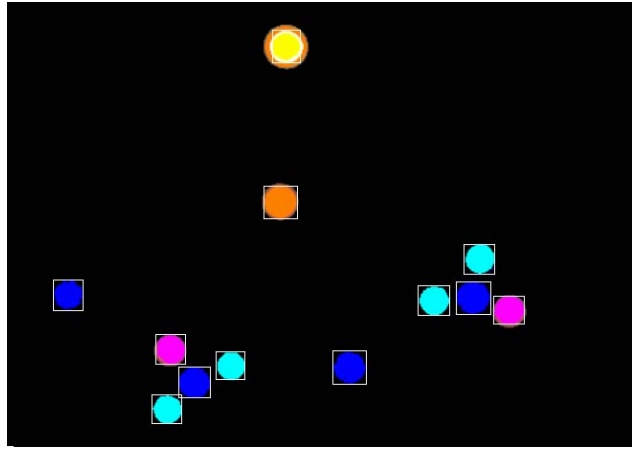


Figure 11.b: Entity classified image

Description

An **Object** is a Center Entity that may be surrounded by one or more Peripheral Entities. Its properties are stored in Structure FOBJ with the following members:-

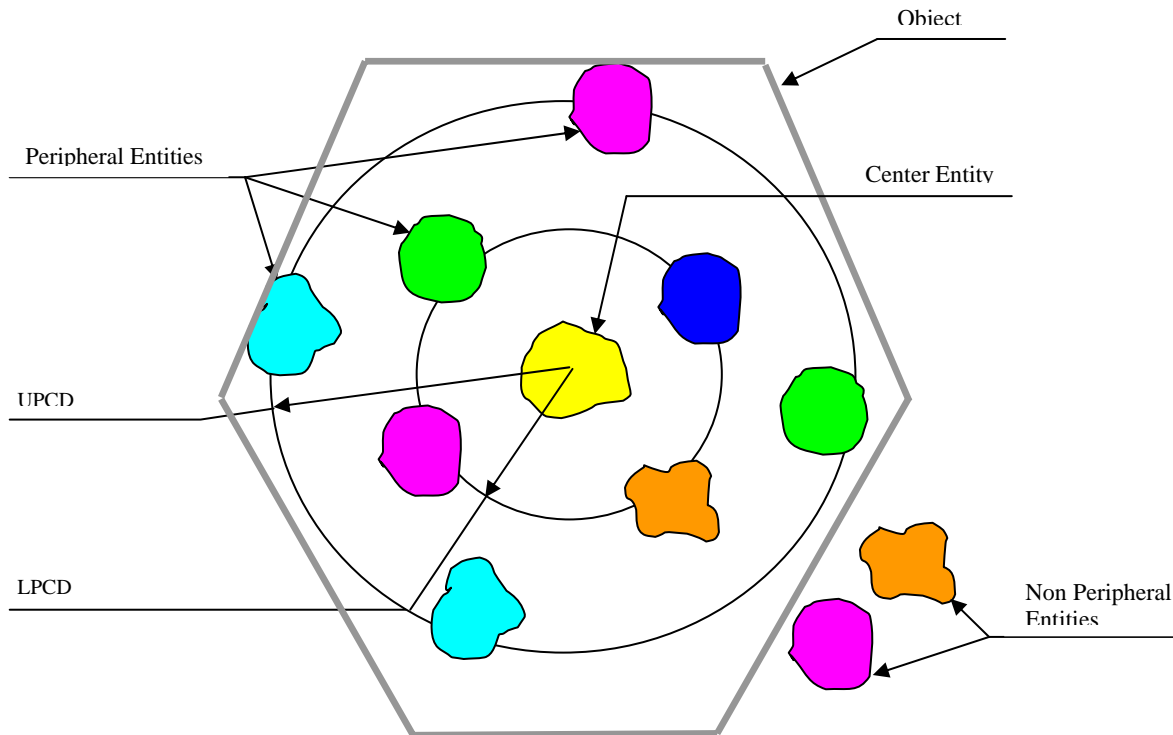
- *r_dp*-Center of Gravity of Peripheral Entities
- *r_cp*-Center of Gravity of Center Entity
- *r_no*-Object number

Peripheral Entities, if any, are related to the Center Entity by the distance of their centers. If this distance is within the specified limits, the peripheral Entity is a member of the Object. The *type* of Object is determined by the color of its Center Entity, thus an object with a Blue colored Center may be called a 'Friendly robot' many of which may be present in the image.

GetObject() starts with the 1st Center Entity color(object type) in *cc[]* and examines each valid Entity of that Color, in *coli[][]*, one at a time. For the selected Center Entity color, it selects the corresponding Peripheral Entity color array in *fc[][]*. Therein, It selects the 1st Peripheral Entity color and examines each valid Entity of that color, in *coli[][]*, one at a time. Any Peripheral Entity with a distance from the Center Entity within the limits set by *lpcd* & *upcd* is included in *rdb[]* as a member Entity, the number of elements(*rc2*) in *rdb[]* is updated as is *nob[]*. It then goes to the next Peripheral Entity color and repeats the process. After all the Peripheral Entity colors for the selected Center Entity are processed, the process repeats with the next Center Entity. After all the Center Entities of a color are done, the program skips to the next Center Entity Color & repeats the process.

Once a Peripheral Entity becomes a member to an Object, it is not marked. In other words the *upcd* circle must be within the Object to prevent a Peripheral Entity from becoming a member to more than one Object.

Graphical Representation



Inputs - Setting

Pass the address of a structure variable of type ROBB, with the following members, to GetObject().

- ncl - Number of Center Entity colors(Object types)
- cc[i] - Center Entity color code for Object type(i). $0 \leq i < ncl$.
- np[i] - # of Peripheral Entity colors(excl black) for Object type(i). $0 \leq i < ncl$
- nob[i] - # of objects expected, of Object type(i). $0 \leq i < ncl$
- fc[i][j]- Peripheral Entity color codes for Object type(i). ($0 \leq i < ncl, 0 \leq j < np[i]$).
- Dfile - Pointer to a file name string to save Object details to the disk
- Dprn - Save(1)/No Save(0) Object details to the disk
- Lpcd - lower limit of Center to Peripheral Entity distance
- Upcd - upper limit of Center to Peripheral Entity distance

Inputs - GetEntity() Outputs

Pass the address of a structure variable of type IMAGE, with the following members, to GetObject().

ncc = # of color codes defined

- dpt[i] - Number of Entities generated of color code(i+1). $0 \leq i < ncc$.
- Coli[i][j] - Array of Entities ($0 \leq i < ncc, 0 \leq j < dpt[i]$), each depth element(i) corresponds to color code(i+1) and contains the start address of the first width element(j). # of width elements = # of entities of that color code(dpt[i]), each element containing one Entity.

Return Values

GetObject() returns values in a structure variable of type FOBD, whose address is passed to it.

- nob[i] - # of objects generated, of Object type(i). depth $\geq ncl$. Declared outside the function w/ a pointer to it passed in the structure variable.
- orn[i] - 1st element# in rdb[] of Object type(i). depth $\geq ncl$. Declared outside the function w/ a pointer to it passed in the structure variable.
- rdb - 1D array of Object structures
- rc2 - Element count of rdb[]

Object Numbering

Object Classification starts with Object type and segment #(i=0) and proceeds upto i=ncl-1.
 Object numbers, given during Classification, are given as:-
 Object number = Segment value + offset.

Segment value is the highest Object # in the preceding segment.

If np[i]=0 the offset is a sequential numbers from 1 onwards.

If np[i]>0, the offset is obtained by decoding the Peripheral Entities of the Object as shown below.

Entity Details

Entity Color	Color Name	R	G	B	Color Code	Entity Position
	Black	0	0	0	0	Peripheral Entity
	Cyan	0	255	255	2	Peripheral Entity
	Magenta	255	0	255	3	Peripheral Entity
	Blue	0	0	255	1	Friendly Robot Center Entity
	Yellow	255	255	0	4	Enemy Robot Center Entity
	Orange	255	128	0	5	Ball Center Entity

Object Details

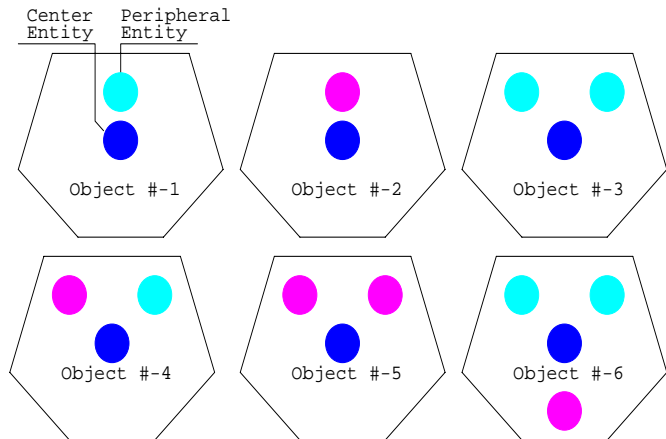
Object Type (*ncl=3)	Center Entity Color Code(*cc)	#Peripheral Entities Excluding black(*np)	Peripheral Entity Color Code(*fc)	# Objects (*nob)
Friendly Robot(i=0)	cc[i]=1	np[i]=2	fc[i]=[2,3]**	nob[i]=8
Enemy Robot(i=1)	cc[i]=4	np[i]=0	-	nob[i]=8
Ball(i=2)	cc[i]=5	np[i]=0	-	nob[i]=1

*Are inputs to GetColor() see 'Setting Inputs'

**Peripheral Entity Color Codes must start from 2 and they must be sequential ie there should be no gaps between two successive codes

Object Number

Peripheral Entity Color Code Combination	Total	Object# (Total-1)
0,0,0	0	-1
0,0,2	2	1
0,0,3	3	2
0,2,2	4	3
0,3,2	5	4
2,2,2	6	5
0,3,3	6	5
3,2,2	7	6
3,3,2	8	7
3,3,3	9	8



As seen the number of positive & unique Object Numbers=Number of Objects(No)

$No = (np+1)^2 - 1$

Object Orientation

The Center of Gravity of the Peripheral Entities can be used to determine the orientation of the Object(Angle wrt +ive x axis)

Angle= $Tan^{-1}(dy/dx)$

Where:

dy=Difference in Center of Gravity 'y' co-ordinate of Center and Peripheral Entities

dx=Difference in Center of Gravity 'x' co-ordinate of Center and Peripheral Entities

Error Detection

Errors occur under the following conditions:-

- 1>The number of Peripheral Entities an object of type 'i' is $>np[i]+1$
- 2>The number of Peripheral Entities an object of type 'i' is $=0$ and $np[i]>0$
- 3>The number of Objects of type 'i' found(in the return structure 'nob[i]') are greater than expected(in the input structure 'nob[i]')
- 4>Duplicate Object numbers exist
- 5>The Peripheral Entities have been wrongly detected resulting in non sequential Object numbers