

Settings

Key Field Internal Edit Control

Buttons

*Note: See DlgFunc.doc for description of 'Buttons', 'Key Field' and 'Internal Edit Control' boxes.

Name

A 14 char wide field, in the Settings file used by the 'Find' button in conjunction with the contents of the 'Search' box, to find or go to records in the file(see DlgFunc.doc).

Color Classification Settings

'member' objects are members of the 'PSET' structure. See ImgFunc.doc for details. Each row in Table 1(fig 2) represents one of 'N' color segments in the raw image.

- **Foregnd**

3 digit integer values for 'Foreground' settings:-

- Color Code-Color code. Range:1 to N+2, taken into the 'eclr[1]' member.
- Intensity -Intensity limit. Range:0 to 255, taken into the 'eint[1]' member.

- **Backgnd**

3 digit integer values for the 'Background' settings:-

- Color Code-Color code. Range:1 to N+2, taken into the 'eclr[0]' member.
- Intensity - Intensity limit. Range:0 to 255, taken into the 'eint[0]' member.

- **File Name and Opn**

- I/P

Opn and File name are taken into the 'fop[4]' and 'fnm[4]' members *resp.* The 'SnapShot' menu option saves an image into the file specified. The file is used when the 'fop[4]' box=0.

- O/P

Opn and File name are taken into the 'fop[0]' and 'fnm[0]' members *resp.*

- **Scale Factor**-Taken into the 'scale[0]' member.

- **Segment Settings**

- Upper and Lower Threshold

These are upper and lower threshold values of **Hue**, **Saturation** and **Intensity** for each color segment. Taken into 'H[]', 'S[]' & 'I[]' members *resp.* **Hue & Sat** vary from 0 to 240 and **Int** from 0 to 255.

- Color

Red, Green & Blue components(0 to 255) for each color segment. Codes of the two additional colors may be used for the 'Backgnd' & 'Foregnd' settings. Taken in the 'r[]', 'g[]' and 'b[]' members *resp.*

Entity Classification Settings

'member' objects are members of the 'PSET' structure. See ImgFunc.doc for details. Each row in Table 1(fig 2) represents one of 'N' color segments & entity types in the raw image.

- **File Name and Opn**

- Smoothened -Opn and File name are taken into the 'fop[3]' and 'fnm[3]' members *resp.*
- O/P -Opn and File name are taken into the 'fop[1]' and 'fnm[1]' members *resp.*

- **Smoothen Image**-Taken into the 'smth[0]' member

- **Entity Area**

These are upper & lower areas(in pixels, 0 to 999) of each entity type. Taken into the 'pcnt[]' member

Object Classification Settings

'member' objects are members of the 'PSET' structure. See ImgFunc.doc for details.

- **Object Numbers**

The number of objects, of each type, expected in the image. Taken in the 'nob[0..2]' member for error detection.

- **Center Color**

Color code(1 to N) of the center entity of each object type, taken in the 'nob[3..5]' member.

- **File Name and Opn**-Opn and File name are taken into the 'fop[2]' and 'fnm[2]' members *resp.*

- **Peripheral Entity Settings**

- PCD(Hi/Lo)

Limits of radial distance between the cg of an object's center entity a peripheral entity, for the peripheral entity to be considered a part of the object. The Hi and Lo limits are taken in the 'pcd[1]' & 'pcd[0]' members *resp.*

- Number off

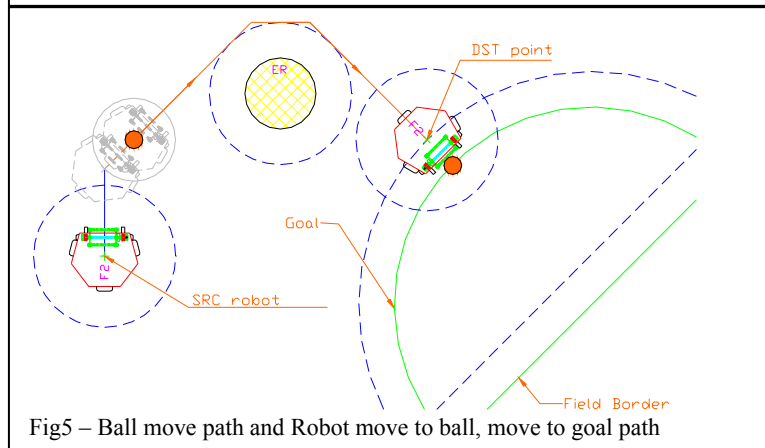
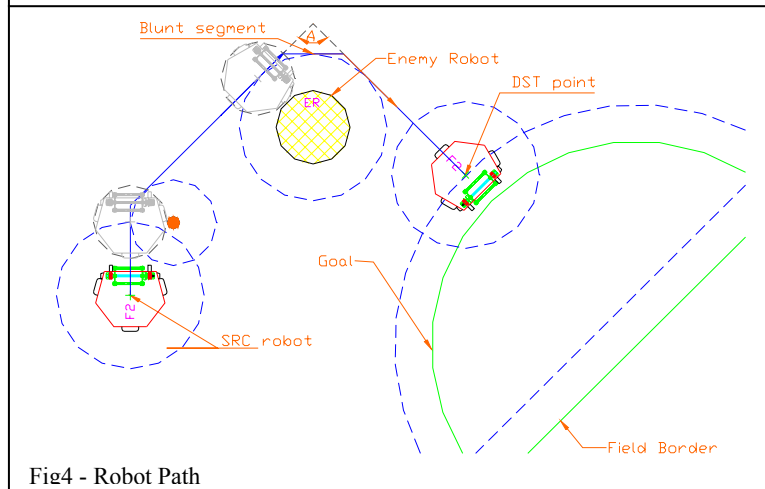
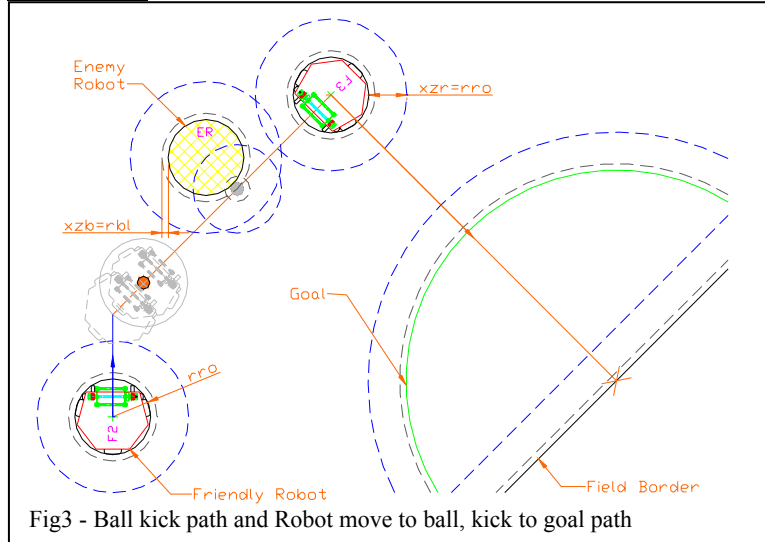
Number of Peripheral Entities and Peripheral Entity colors(incl black), taken into the 'np[0][[]]' member.

Path Plan Settings

Objects referred to as 'member', are members of the 'PSET' structure, passed as an i/p to GetEntity().
The File Names mentioned here have a width of 13.

After 'Object Classification' by either GetObject() or aread() depending on the 'fop[5]' option & before Path Planning, mkrb() is called where the Enemy goal is added, objects are given a radius & the objects are skew corrected

Path Types



A Cumulative Path is a series of end to end connected Path(line) segments from the center of the source object to a destination point. Paths can be planned on a static or dynamic field. The former requires the 'Frame Processing' menu option, the latter requires the 'Game Processing' menu option. The i/p for both are specified by the 'Fop[6]' option below.

There are basic and composite paths.

Basic paths

- **Ball kick path** to goal is shown in red(fig 3).
- **Ball move path** to goal is shown in red(fig 5).
- **Robot path** of robot f2, shown in blue to ball(fig 3,5), to DST point(fig 4).

Composite paths

- **Robot move to ball, kick to goal path** - fig 3
A friendly robot is selected as the source & its composite path('Ball kick path' to goal and robot path to ball) is made by connecting the above two paths with a connecting path(red dotted). The process is repeated for all friendly robots & the one with smallest composite path selected. The dotted red line connecting the two basic paths is the **kicker on segment** where the robot turns on the kicknd moves towards the ball.
- **Robot move to ball, move to goal path** - fig 5
A friendly robot is selected as the source & its composite path('Ball move, path' to goal and robot path to ball) is made by connecting the above two paths with a connecting path(red dotted). The process is repeated for all friendly robots & the one with smallest composite path selected.

- DST - Field Object # at whose center a Cumulative Path terminates. Taken into 'dstc[2]' member
- DST Point - Co-ordinate(x,y) of point at which a Cumulative Path terminates. Taken into 'dstc[0,1]' member.
- Ext - Taken into 'dstc[4]' member. May be 1 or 0
 - 1 - The Cumulative Path terminates exactly as specified.
 - 0 - The Cumulative Path termination may be near the point specified.
- Opn
Static Path Plan Option. Taken into 'dstc[5]' member:-
 - 0 - Move robots on **custom Paths**, specified in 'Acad i/p File'. SRC, DST, 'DST point' and Exact are unused.
 - 1 - **Ball kick path** to the goal. Kicker robot is SRC. DST and 'DST point' are unused, no RF
 - 2 - **Ball move path** to goal. Robot is SRC. DST and 'DST point' are unused, no RF.
 - 3 - **Robot path** to 'DST point'. Robot is SRC. DST is unused
 - 4 - **Robot path** to 'DST'. Robot is SRC. 'DST point' is unused.
 - 5 - **Robot move to ball, kick to goal**. SRC, DST, 'DST point' and Exact are unused.
 - 6 - **Robot move to ball, move to goal**. SRC, DST, 'DST point' and Exact are unused.
 - 7 - At the location specified in the 'Acad o/p File' edit box, generates an autocad script file(fld0.scr) for the skew corrected field & objects, superimposed on the original field(in grey) w/o objects. This option ignores the 'Opn' box of the 'Acad o/p File' box.
- Accl – Use 'Acad i/p file' ?. Taken into 'fop[5]' member. May be 0 or 1
 - 1 - Yes. The 1st operation before Static path planning is 'Object Classification' with input from 'Acad i/p File'.
 - 0 - No. The 1st operation before Static or Dynamic path planning is 'Color Classification' with input depending on the 'Opn' option of the 'I/P File Name' box in 'Color Classification Settings', when 1 it is taken from the image file specified & when 0, from the camera.
- Acad i/p File
When 'Fop[5]=1, the autocad i/p for 'Object Classification' is taken from a file whose name and location is mentioned in this edit box and taken into the 'fnm[5]' member. It is created as follows(see fig6):-
 - 1>Create a new autocad drawing.
 - 2>Using the 'XATTACH' command insert the field(FIELD), friendly robots(F1 to F6), enemy robots(ER), ball(BALL) and enemy goal(GOAL), external drawing references(XREFs) into it.
 - 3>Rotate the friendly robots as required and make a cumulative path for one or more friendly robots with end to end connected segments of the following colors(consecutive segments may have the same color):-
 - Green - Forward motion
 - Red - Reverse motion
 - Blue - Forward motion with kicker on
 - 4>Closed paths(end point of a path segment is coincident on the start point of another path segment) can be made, however the segment selected by the robot from the point of coincidence is unknown. If the selection is important, displace the end of the last segment by 0.0001mm(approx) from the start of the first.
 - 5>Use the 'DBLIST' command and copy-paste its output into an MS-WORD document saved as an MS-DOS text file. Enter the name and location of this file into the 'Acad i/p File' edit box.

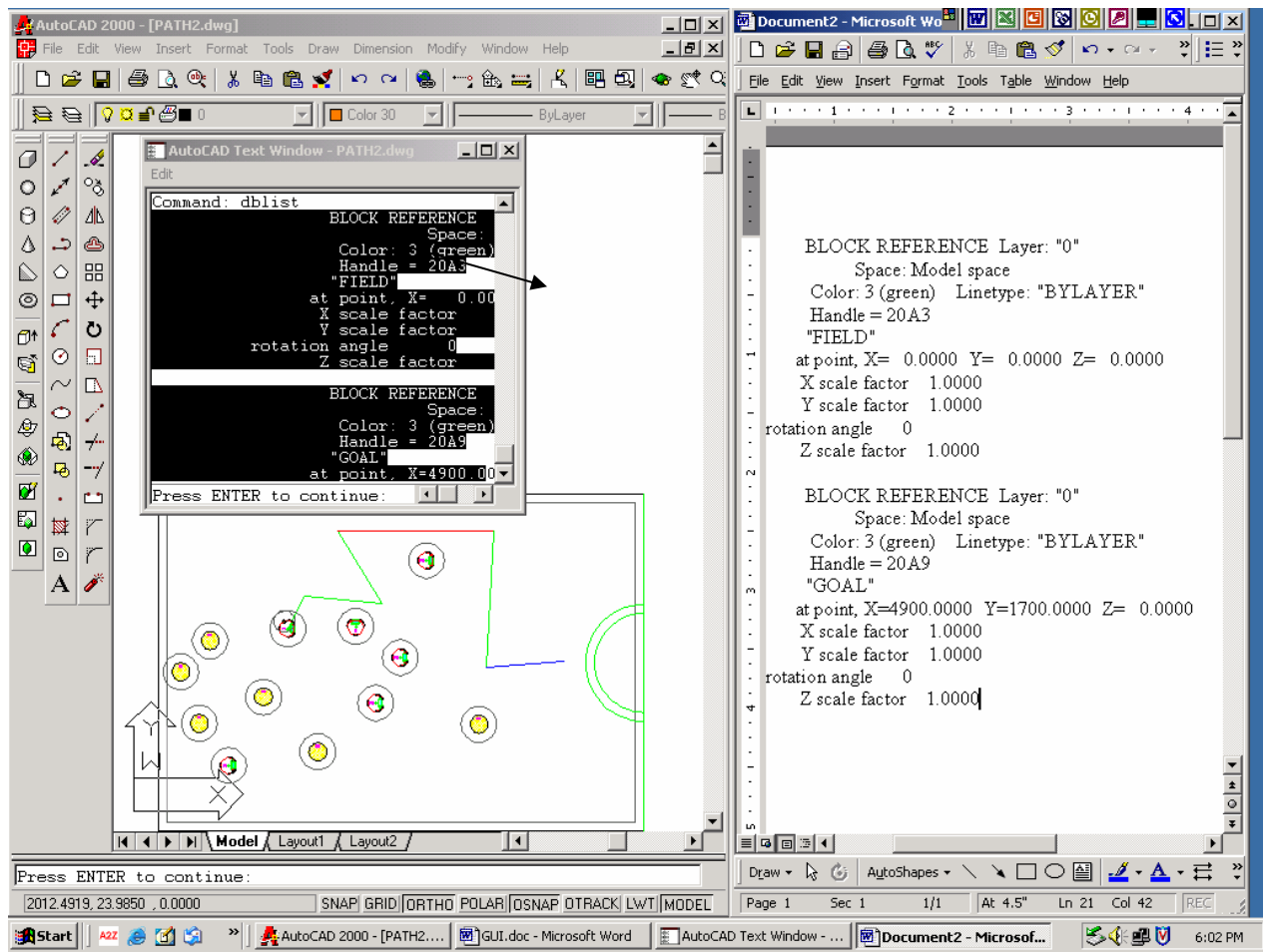


Fig6 – Autocad i/p file generation

- **Radii**

- **Goal** - Radius of Goal in 'mm'. Taken into 'rad[2]' member

- **Ball**

Fig 3 shows a 'Ball kick path'. After making the source robot(kicker) transparent, an exclusion zone(xzb) equal to the **Ball radius** is constructed around field objects, to prevent collisions with the ball, by ensuring that path segments do not intersect the exclusion zone(xzb). If collision is observed, the ball radius(in mm), taken into the 'rad[1]' member, may be increased.

- **Robot**

Fig 4 shows a 'Robot path' moving the source(SRC) robot 'F0' to destination(DST) point. An exclusion zone(xzr) equal to the **Robot radius** is constructed around field objects, to prevent collisions with the SRC object, by ensuring that path segments do not intersect the exclusion zone(xzr). If collision is observed, the robot radius(in mm), taken into the 'rad[3]', member may be increased.

- **Rob PCR**

When a robot rotates, the inner and outer rim of its wheels slip, with the outer rim rotating faster than the inner. The plane passing through the wheel, between the inner and outer rim, doesn't slip and is used to calculate the circumferential displacement for a given angular displacement. A circle tangent to this plane with its centre at the robot center, is the pitch circle and its radius the **Pitch Circle Radius(PCR)**. PCR in 'mm' is taken into the 'rad[0]' member.

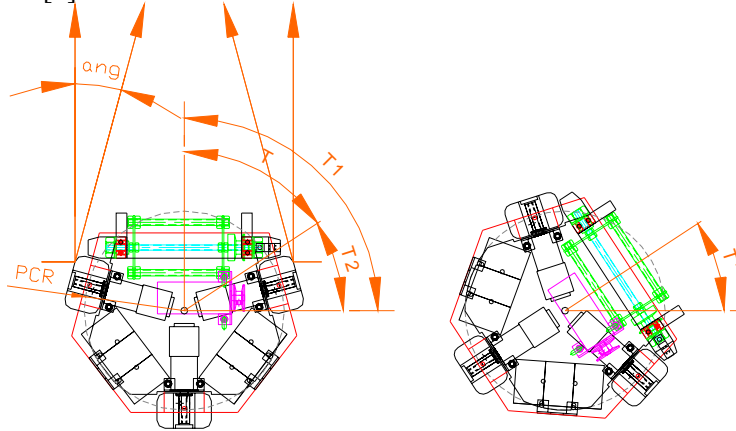


Fig7a Robot at position 1

Fig7b Robot at position 2

The Circumferential displacement 'D' required to rotate the robot from position 1 to position 2 is:-

$$C=2\pi*PCR$$

-- Pitch Circle Circumference

$$D=2\pi*PCR*(T/2\pi)$$

$$D=PCR*T$$

-- See doc on MAKE(), generation of serial Tx file

Where 'T' is calculated as follows:-

$$\tan(T)=\tan(T2-T1)=[\tan(T2)-\tan(T1)]/[1+\tan(T2)*\tan(T1)]$$

Let:-

$$\tan(T2)=m2, \tan(T1)=m1, \tan(T)=m$$

Then:-

$$m=[m2-m1]/[1+m2*m1]$$

and

$$T=\tan^{-1}(m)$$

- **Com Port Settings**

- **Port Number** - Serial port#: 1 for COM1. "COM1" string taken into 'comp[0]' member
- **Baud Rate** - Serial port baud rate. Taken into 'baud[0]' member.

- **General Settings**

- **Rec Space in FPGA**

A record of data sent by RF to the robot processor(FPGA) is a control word and a path length.

The number of such records that can be sent to the FPGA depends on how the FPGA has been configured and is currently set to (x=28). The Game Processing and Static Path Plan menu options transmit 'x' and 1 records *resp.* This setting must be set to 'x' and is taken into the 'fpga[0]' member.

- Path Len to FPGA is Float

The path length constituent of a record to data sent to the robot FPGA may be a 16 bit integer or a 32 bit float value(real4 format), depending on the configuration of the FPGA. The current setting must be(0) for integer and is taken into the 'fpga[1]' member.

- Blunt Angle

If the angle(A), between path segments of a **robot path** or **ball move path**, is less than this setting, the path is blunted as shown in Fig 4. The process of blunting continues recursively until the angles between the segments satisfy the constraint. $\text{Cos}(\text{Blunt Angle}/2)$ is taken into the 'bfc[0]' member.

- Path Hysteresis

Below are two 'basic' paths for a robot to a destination. The shorter path has been determined, the longer path is being determined. Segments of this path(grey) are ignored once its length exceeds the shorter path by a percentage specified in the 'Path Hysteresis' edit box. Also applies to 'composite' paths, but not to the 'basic' paths within them. $\text{Path Hysteresis}/100$ is taken into the 'acc[0]' member.

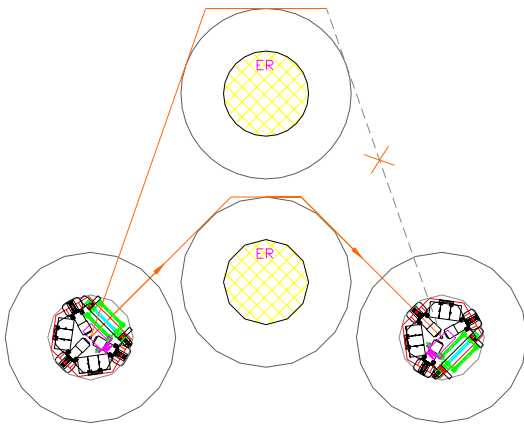


Fig9 - Path Hysteresis

- Runup for Kicker

There are two ways of kicking & moving a ball. a> can be used only when the robot has a ball gripper.

a>Robot rotates, the **ball moves** with it, to the direction required.

b>**Ball is stationary**. The robot moves to position itself such that the ball is in the direction required (fig 3,5). If the robot has a rotatory kicker, the kicker takes time to accelerate and may require the kicker on segment (fig 3) to be longer, the excess length is entered in this box and taken into the 'otx[0]' member.

- Front Wheel Angle

When a robot moves in a straight line(translate), the angle of the front wheels('ang' in degrees), with the vertical, causes them to slip. The forward displacement(v1) of the robot is thus a component of the circumferential displacement of the forward wheels(v2) and is given as(see Fig 7):-

The factor 0.9659258(below) is taken into the 'fc0[0]' member.

$$v1=v2*\text{Cos}(\text{ang})$$

When fc0[0]=15 degrees:-

$$v1=v2*0.9659258$$

- Min Distance

If robot translation(mm) is less than this, translation doesn't occur. Taken into the 'min[0]' member.

- Min Angle

If robot rotation(degrees) is less than this, rotation doesn't occur. Taken into the 'min[1]' member.

- Acad o/p File & Opn

Script files for Autocad may be generated for the field and field objects(ball, friendly & enemy robots, goal). Overlaid on this are Tangents and Paths depending on the entry(0 to 3) in the 'Opn' edit box, taken into the 'fop[6]' member. 'Opn' is used when the 'Game Processing' menu option is selected or 'Static Path Plan' menu option is selected with the 'Static Path Plan Option'<7. When this option=7, a field & field objects file 'Fld0.scr' is generated at the path specified in 'Acad o/p File', irrespective of 'Opn'.

Opn

0 - No file generated

1 - Generate a file for all Tangents and a file for each Cumulative path

2 - Generate a file for all Tangents and a file for all Cumulative paths

3 - Generate a file for all Tangents, a file for each Cumulative path and a file for all Cumulative paths

Acad o/p File

The 'Acad o/p File' edit box requires a path for the file and is taken into the 'fnm[6]' member. File names are generated internally and must not be included in the entry. File naming convention is as follows:-

- Tangent File Name

1>"TanRrS" -Robot path + field & field objects (fig 4)

2>"TanRbS" -Ball move path + field & field objects (fig 5)

3>"TanBS" -Ball kick path + field & field objects (fig 3)

Where S is: File number(S>=0)

- Path File Name

1>"PthRrN_S" -Robot path + field & field objects(fig 4)

2>"PthRbN_S" -Ball move path + field & field objects (5)

3>"PthBN_S" -Ball kick path + field & field objects (fig 3)

- Field & Field Objects

1>"Fld0" -Field & Field Objects generated when 'Static Path Plan Opn'=7 & 'Static Path Plan' menu option is selected. See 'Opn' under 'Path Plan Settings' for details.

Where N is: Kicker robot #(file type 2> and 3>), Robot #(file type 1>) and

S is: File number(starts from 0 for every value of N).

- Field Settings

- Enemy Side

Enter 'L'(Left Hand Side) or 'R'(Right Hand Side). Taken into the 'esd' member.

- Real Field Dimensions

Dimensions of the real world field in 'mm'. Taken into the 'fxy[]' member. fxy[0]→length(x), fxy[1]→width(y)

- Image Field corners

'Spatial' co-ordinates, in pixels, of the playing field corners in the bitmap image, as seen in MS-Paint, converted to 'Cartesian' -see 'GetColor.doc'.

Co-ordinates entered are:- Top left(TL), top right(TR), bottom right(BR) and bottom left(BL).

Taken into the 'crn[]' member . crn[0]→TL(x), crn[1]→TL(y), crn[2]→TR(x), crn[3]→TR(y), crn[4]→BR(x), crn[5]→BR(y), crn[6]→BL(x), crn[7]→BL(y)

- Acad Field corners

'Cartesian' co-ordinates in 'mm', of the playing field corners in the AutoCad i/p file-see 'GetColor.doc'.

Co-ordinates entered are:-Top left(TL), top right(TR), bottom right(BR) and bottom left(BL).

Taken into the 'arn[]' member . arn[0]→TL(x), arn[1]→TL(y), arn[2]→TR(x), arn[3]→TR(y), arn[4]→BR(x), arn[5]→BR(y), arn[6]→BL(x), arn[7]→BL(y)