

```

outlier -----> { vectorValue
                  { line
                  { pNext

lineVectors ----> { time
                  { vectorX
                  { vectorY
                  { vectorZ

define THRESHOLDX_LEFT 0.00096087 - 3* 0.38875666
define THRESHOLDX_RIGHT 0.00096087 + 3* 0.38875666

define THRESHOLDY_LEFT 0.05525659 - 3* 0.61937128
define THRESHOLDY_RIGHT 0.05525659 + 3* 0.61937128

define THRESHOLDZ_LEFT 0.0352192 - 3* 0.4300345
define THRESHOLDZ_RIGHT 0.0352192 + 3* 0.4300345

o-----o
| main |
o-----o
*
paths[15][7] = "dws_1/", "dws_2/", "dws_11..."

pFileOutliers = opening the file (in append mod)
if(pFileOutliers == NULL)
    print "Error when opening the file: review_outliers"
    exit

iMovement = 0
while (iMovement < 15)
    iSub = 1
    while (iSub ≤ 24)
        url = paths[iMovement] + "sub_" + iSub + ".csv"
        pFileSub = opening the file (in read mod) from the url

        if(pFileSub == NULL)
            print "Error when opening the file: " + url
            exit

        line = get the line of pFileSub

        nbLine, time, isMissingTime = 0
        previousTime = -1
        nbOutliersX, nbOutliersY, nbOutliersZ = 0

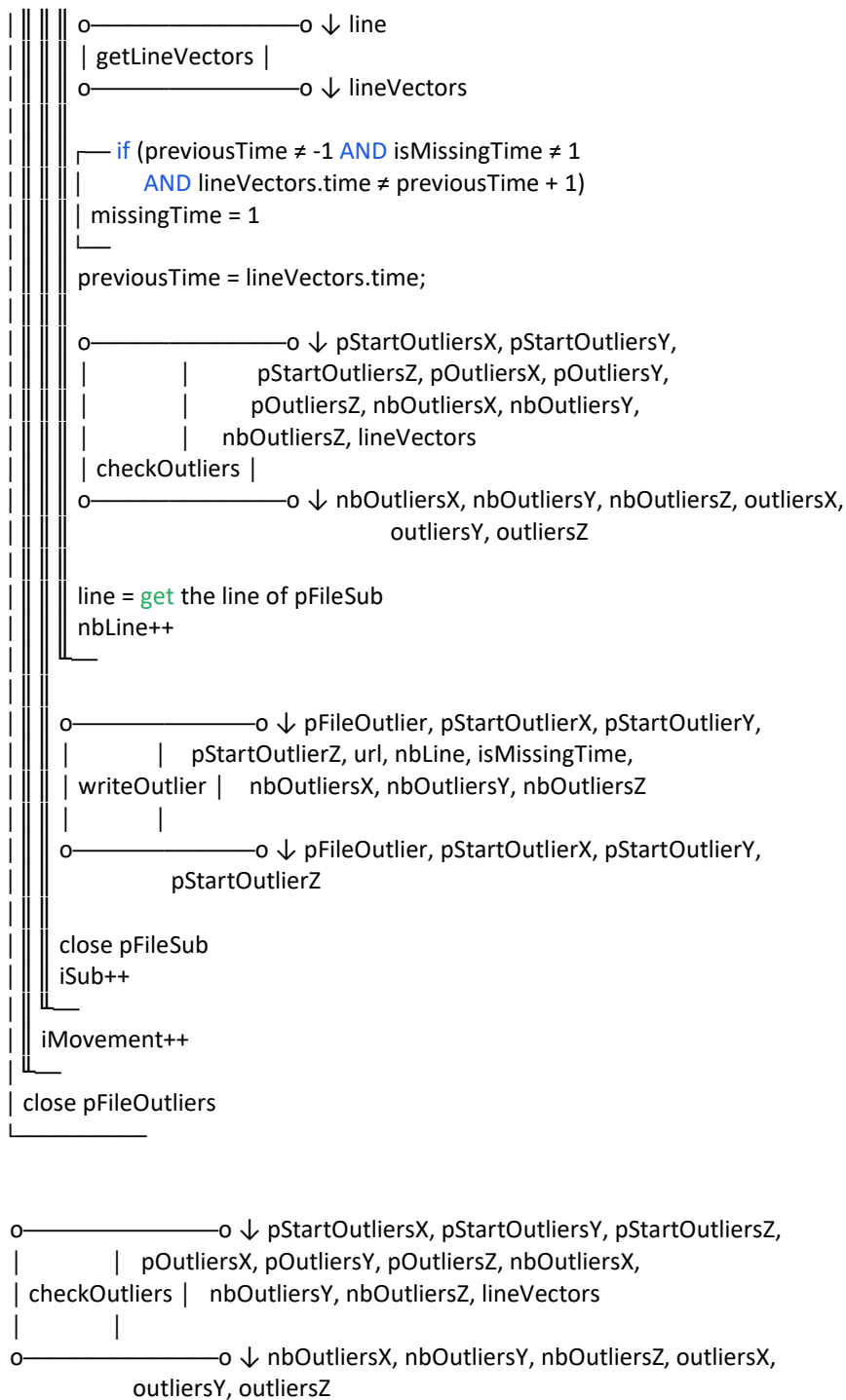
        pOutliersX = pStartOutliersX = reserve memory (struct outlier)
        pOutliersY = pStartOutliersY = reserve memory (struct outlier)
        pOutliersZ = pStartOutliersZ = reserve memory (struct outlier)

        line = getLine of pFileSub // skip the header

        while (line.getline ≠ null)

```

## DA Phase 1



```

|  *
|  ┌─ if (lineVectors.vectorX < THRESHOLDX_LEFT
|  |   OR lineVectors.vectorX > THRESHOLDX_RIGHT)
|  | newOutlier = reserve memory for (struct Outlier)
|  | newOutlier→line = lineVectors.time
|  | newOutlier→vectorValue = lineVectors.vectorX
|  | newOutlier→next = NULL
|  |
|  | ┌─ if (pStartOutlierX == NULL)
|  | | pOutlierX = pStartOutlierX = newOutlier
|  | | └─ else
|  | |   pOutlierX→next = newOutlier
|  | |   pOutlierX = pOutlierX→next
|  | └─
|  | nbOutlierX++
|  └─
|
|  ┌─ if (lineVectors.vectorY < THRESHOLDY_LEFT
|  |   OR lineVectors.vectorY > THRESHOLDY_RIGHT)
|  | newOutlier = reserve memory for (struct Outlier)
|  | newOutlier→line = lineVectors.time
|  | newOutlier→vectorValue = lineVectors.vectorY
|  | newOutlier→next = NULL
|  |
|  | ┌─ if (pStartOutlierY == NULL)
|  | | pOutlierY = pStartOutlierY = newOutlier
|  | | └─ else
|  | |   pOutlierY→next = newOutlier
|  | |   pOutlierY = pOutlierY→next
|  | └─
|  | nbOutlierY++
|  └─
|
|  ┌─ if (lineVectors.vectorZ < THRESHOLDZ_LEFT
|  |   OR lineVectors.vectorZ > THRESHOLDZ_RIGHT)
|  | newOutlier = reserve memory for (struct Outlier)
|  | newOutlier→line = lineVectors.time
|  | newOutlier→vectorValue = lineVectors.vectorZ
|  | newOutlier→next = NULL
|  |
|  | ┌─ if (pStartOutlierX == NULL)
|  | | pOutlierZ = pStartOutlierZ = newOutlier
|  | | └─ else
|  | |   pOutlierZ→next = newOutlier
|  | |   pOutlierZ = pOutlierZ→next
|  | └─
|  | nbOutlierZ++
|  └─
|
└─

```

## DA Phase 1

```
o-----o ↓ pFileOutlier, pStartOutlierX, pStartOutlierY,  
|       | pStartOutlierZ, url, nbLine, isMissingTime,  
| writeOutlier | nbOutliersX, nbOutliersY, nbOutliersZ  
|       |  
o-----o ↓ pFileOutlier, pStartOutlierX, pStartOutlierY,  
          pStartOutlierZ  
*  
| write in pFileOutlier : url + numLine + isMissingTime + nbOutliersX  
|  
| outlierX = pStartOutlierX  
| | while(outlierX ≠ null)  
| | | write in pFileOutlier : outlierX→line + outlierX→vectorValue  
| | | nextX = outlierX→pNext  
| | | free(outlierX)  
| | | outlierX = nextX  
| |  
| write in pFileOutlier : nbOutliersY  
| outlierY = pStartOutlierY  
| | while(outlierY ≠ null)  
| | | write in pFileOutlier : outlierY→line + outlierY→vectorValue  
| | | nextY = outlierY→pNext  
| | | free(outlierY)  
| | | outlierY = nextY  
| |  
| write in pFileOutlier : nbOutliersZ  
| outlierZ = pStartOutlierZ  
| | while(outlierZ ≠ null)  
| | | write in pFileOutlier : outlierZ→line + outlierZ→vectorValue  
| | | nextZ = outlierZ→pNext  
| | | free(outlierZ)  
| | | outlierZ = nextZ  
| |  
| write in pFileOutlier : "\n"
```