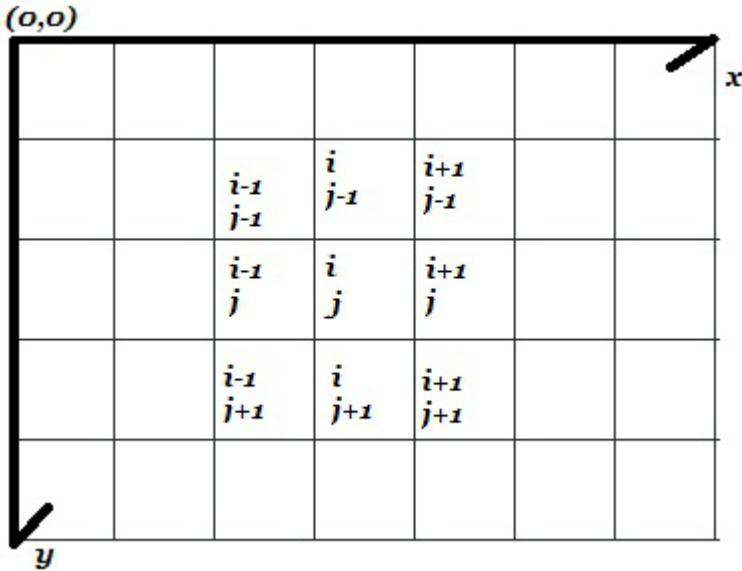


Simple Random Walk

May 27, 2013

Review of some related concepts

The pixel coordinate



For a random walk, first initialize the (x, y) location

```
int x, y;
```

And update the current location by $x = x + \text{update}$

The `update` is generated by random number generator

For example, in C++ , the `rand()% a`; generate a random number between $[0, a]$

Simple Random walk 1

```
void randomMove1(int &x, int &y){
    int choice = int( rand() % 4 );
    if      ( choice == 0 ){ x++; }
    else if ( choice == 1 ){ x--; }
    else if ( choice == 2 ){ y++; }
    else           { y--; }
}
```

This method is simple but can only move in 4 direction

Simple Random walk 2

To extend the moving direction from 4 to 8 , just use a larger range of random number and assign the direction in each case.

```
void randomMove2a(int &x, int &y){
    int choice = int( rand() % 8 );
    switch( choice ){
        case 0 : x--;y--; break;
        case 1 :      y--; break;
        case 2 : x++;y--; break;
        case 3 : x--      ; break;
        case 4 :      ; break;
        case 5 : x++      ; break;
        case 6 : x--;y++; break;
        case 7 :      y++; break;
        case 8 : x++;y++; break;
    }
}
```

This kind of case structure seems good enough, but further improvement can be made by using combination

```
void randomMove2b(int &x, int &y){
    int step_x = int( rand() % 3 ) - 1;
    int step_y = int( rand() % 3 ) - 1;
    x += step_x;
    y += step_y;
}
```

The statement `int(rand()%3)` generate integers 0,1,2 and thus `int(rand()%3) - 1 = {-1,0,+1}`

The statement `x += step_x;` is the update of the coordinate in x

Simple Random Walk with arbitrary step size

In the previous methods , the update step size is limited to 1, but it can be generalized to any step size.

–END–