

MAEOCS: Mobile Application for Easy Orientation in Confined Spaces

Carlos I. Gaitán, *Estudiante, Escuela Colombiana de Ingeniería Julio Garavito*, y
Edward H. Jiménez, *Estudiante, Escuela Colombiana de Ingeniería Julio Garavito*.

Abstract—Esta investigación es pionera en Escuela Colombiana de Ingeniería Julio Garavito hablando a cerca de georeferenciación usando dispositivos móviles. El alcance de la aplicación es la mejora en la calidad de vida de las personas, reduciendo tiempos y malentendidos.

Index Terms—Android, Java, GPS.

I. INTRODUCCIÓN

Hoy en día es fácil evidenciar cómo los planteles públicos y privados van en crecimiento, ampliando sus zonas en incluso dispersadoras en diferentes puntos de un mismo sector, por ello las personas que son nuevas en estos ambientes, o que poseen mala memoria, alguna discapacidad cognitiva, o tenga problemas para comunicarse a otros, lo cual les impide ubicarse de una forma fácil y se ven afectadas cuando pierden el tiempo mientras miran mapas que son poco claros.

Teniendo en cuenta que en esta era tecnológica, los dispositivos móviles cuentan con una capacidad de procesamiento que día a día aumenta. Las limitaciones que se tenía para resolver problemas de la vida cotidiana a través de un computador se hacen cada vez menos. El crecimiento en tecnología a impulsado un nuevo modelo de desarrollo de aplicaciones para dispositivos móviles. Se han desarrollado frameworks y arquitecturas que facilitan el desarrollo e interacción con distintas tecnologías y estas permiten llevar las ventajas de la computación a la palma de la mano, permitiendo realizar proceimientos y calculos de casi cualquier complejidad en este tipo de dispositivos, dando solución a problemáticas de muchas índoles.

Como se mencionaba anteriormente, la orientación es un problema que a través de la tecnología móvil ya venido siendo tratado. Actualmente se puede una persona puede saber su posición geográfica usando un dispositivo que utilice GPS y seleccionando un destino se puede solicitar una ruta trazada que indique como llegar de un punto a otro. No obstante este tipo de orientación solo funciona actualmente en espacios abiertos y rutas de transito, dejando de lado lugares que pueden llegar a desorientar a las personas que transitan en estos sitios.

II. PLANTEAMIENTO DE LA SOLUCIÓN

Para dar solución a esta problemática hay muchas posibles soluciones, hablando de la arquitectura de la solución que se desea usar; pero el problema de trazar rutas entre dos caminos si se puede llegar a tratar de

una manera general para cualquier solución. Partiendo del hecho que calcular una ruta entre todas las conexiones puede representarse en modelos abstractos de grafos, que ya han sido parte de investigaciones para solución de problemas.

La teoría de grafos ha dado como resultado de investigaciones una serie de algoritmos que responden el problema de búsqueda de caminos, algoritmos que han sido diseñados para dar solución a distintas estructuras de grafos y a su caracterización. Dentro de los algoritmos se encuentran distintas de estrategias para tratar el problema, lo cual hace a algunos de estos más efectivos en problemas específicos. También se han diseñado algoritmos que usan patrones expertos para agilizar la toma de decisiones.

Para encontrar la ruta entre dos puntos ubicados en un mapa que podría contener bucles en sus caminos es necesario buscar un algoritmo que pueda avanzar de una forma ágil y que busque reducir la complejidad del calculo, reduciendo tiempo y recursos tecnológicos por la cantidad de datos que se tienen. Teniendo en cuenta estos requerimientos, se inició la búsqueda de un algoritmo que use un patrón experto, una heurística que reduzca la complejidad algorítmica y el tiempo de respuesta.

Al concluir el resultado de la investigación fue el usar la estrategia usada por "A*" o "A estrella". Un algoritmo que tiene el poder de reducir la complejidad de una solución a nivel lineal de acuerdo a la calidad del experto que maneje; el éxito de esta estrategia esta basada en la exactitud de su patrón experto, al que se le conoce como **heurística**.

La definición de una heurística es algo que debe hacerse con precaución, ya que si se hace una mala definición de esta, el algoritmo podría tener que calcular todas las posibles caminos antes de dar con la respuesta, o quizás nunca alcanzarla. Para evitar una definición errónea, se debe pensar en que tipo de problema se está tratando y que tipo de avance beneficia más. Teniendo en cuenta que el problema de ubicación que se intenta solucionar busca la menor distancia entre dos puntos en un mismo plano, problema que ha sido solucionado por la *geometría euclidiana*, en la que Euclides define en su primer postulado "Dados dos puntos se puede trazar una y solo una recta que los une". La longitud de la recta es la distancia que se espera recorrer, por lo que se definió esta heurística que se define como la comparación de las sumas entre la distancia anteriormente recorrida con el calculo de la longitud de la recta que une cada punto que tiene posibilidad de ser escogido. Escrito en forma algorítmica esta definición se

escribió así:

```
private static String bestChoice(
HashMap<String, ArrayList<String>> choices, Node pointB){
double min = Double.MAX_VALUE;
String best = null;

for (Entry<String, ArrayList<String>> entry :
choices.entrySet()){
Node n = map.get(entry.getKey());
double expert = n.bestDistance(pointB);

if (expert+entry.getValue().size()<min){
min = expert+entry.getValue().size();
best = n.getId();
}
}
return best;
}
```

Teniendo definida la heurística para el algoritmo, se procede a definir el comportamiento del algoritmo en general, el cual consta de un avance **voraz**, comparando entre las conexiones que se han desarrollado con el avance del algoritmo, expandiendo únicamente los nuevos caminos que se pueden formar al avanzar por la mejor opción visible, conservando los otros caminos que no han sido explorados y abandonando los caminos que llegan a su fin sin encontrar el punto esperado.

```
public ArrayList<String> aStar(Node pointA, Node pointB){

HashMap<String, ArrayList<String>> roads;
roads = new HashMap<String, ArrayList<String>>();
ArrayList<String> road = new ArrayList<String>();

road.add(pointA.getId());
roads.put(pointA.getId(), road);

/**
 * Calculates the roads until the desired point comes out
 */
while (!roads.containsKey(pointB.getId())){

HashMap<String, ArrayList<String>> choices;
choices = getChoices(roads);
String selected = bestChoice(choices, pointB);
if (!roads.containsKey(pointB.getId()))
cleanRoads(roads);
map.get(selected).visit();
roads.put(selected, choices.get(selected));

}

return roads.get(pointB.getId());
}
```

Al terminar la ejecución del algoritmo, se encontrará un camino que une los dos puntos, intentando recorrer la menor distancia posible.

III. CONCLUSIONES

La investigación desarrollada llevó a la implementación de A*, algoritmo que busca desarrollar una búsqueda rápida a través de una heurística basada en geometría, ha permitido dar solución a la problemática de unir dos puntos que se encuentren en el mismo plano, permitiendo así dar un paso más hacia la orientación dentro de los lugares que no se encuentran cubiertos por los GPS actuales.

Al tener acceso a equipamiento tecnológico móvil que tiene la capacidad para ejecutar el algoritmo escrito, permite hacer búsquedas de rutas en cualquier lugar, sin la

necesidad de tener una estación de trabajo acentada. Mas bien permite a un mayor número de usuarios beneficiarse de este tipo de aplicaciones, que tienen como meta mejorar la calidad de vida y reducir la pérdida de tiempo causada por la desorientación. Ayudando no solamente a aquellos que cuentan con todas sus facultades físicas, sino a su vez a aquellos que tienen mayor inconveniente por tener limitaciones físicas o mentales.

REFERENCIAS

- [1] Teoría de Grafos.
http://es.wikipedia.org/wiki/Teoría_de_grafos
- [2] http://en.wikipedia.org/wiki/A*_search_algorithm
- [3] http://es.wikipedia.org/wiki/Geometría_euclídea
- [4] Graph Theory, Keijo Ruohonen