

ALGORITMO DE UBICACIÓN Y CONDUCTA EN UN ENTORNO CONOCIDO CON CONDICIONES INICIALES DESCONOCIDAS

César Augusto Uribe Meneses Est, Jaime Ignacio Marín Vargas Est, Andrés Felipe Pedraza Monsalve Est, Leonardo Arango Baena Est y Carlos Andrés Madrigal González MSc.

GEPAR

Universidad de Antioquia

Medellín, Colombia

Marzo, 2007

Resumen. El problema de determinar la ubicación de una plataforma móvil en ambientes conocidos sin saber la posición inicial de esta, es una tarea altamente demandante. Este documento describe un algoritmo que utiliza matrices tridimensionales, el cual usa adquisición de datos de los nodos del entorno para reconocer la posición del objeto. El algoritmo es aplicado usando un robot que tiene como objetivo encontrar dos pelotas de tenis, una con una ubicación específica y otra en una posición desconocida y luego llevarlas a una lugar objetivo, pero desconociendo el estado inicial del robot.

Índice de Término. Algoritmo, detección de nodos, localizar, matrices tridimensionales, robótica, posicionamiento.

Abstract. A highly demanding task is the problem of identifying a mobile platform's initial location with recognized land characteristics. This document describes an algorithm, which utilizes three-dimensional matrixes elaborated with data obtained from the nodes, to recognize the position of an object. A robot, with an unidentified initial position, was used to apply this algorithm. The robot's main goal was to find two tennis balls, one tennis ball had a determined location and the other one didn't. Finally, after finding the tennis balls, the robot had to carry them to a final spot.

Index Terms. Algorithm, locate, nodes detection, robotic, tridimensional arrays, positioning.

INTRODUCCIÓN

El proceso de ubicación de objetos en entornos conocidos; juega un papel de vital importancia en el diseño de comportamientos para sistemas automáticos, como por ejemplo: las redes cooperativas de micro-robots [1], que son utilizadas en variedad de procesos que implican una actividad de reconocimiento exhaustiva. En este documento se explica y analiza un algoritmo de ubicación para plataformas móviles cuya posición inicial es desconocida, pero se posee información acerca del entorno en el que se encuentra, además de dispositivos de captura de datos para reconocer las condiciones cercanas de la plataforma.

En este desarrollo se realiza una simplificación del entorno y se parte del hecho de que las posibles variaciones que este posee solo implican bifurcaciones del camino, el cual puede generar trayectorias cerradas interconectadas entre sí, además de limitar las condiciones del terreno a situaciones binarias para los dispositivos de captura de datos. Esta particularización, reduce el conjunto de escenarios utilizadas para la comparación e identificación de puntos singulares, agilizando el proceso de ubicación. Es fácil encontrar este tipo de situaciones específicas en actividades robóticas de aplicación didáctica como los concursos y olimpiadas de robots que se realizan en diferentes entes académicos con el fin de impulsar la investigación y el desarrollo en la robótica [2].

Para la aplicación de este algoritmo es necesario que el interesado tenga un conocimiento básico en programación y manejo de arreglos multidimensionales en lenguajes convencionales

como: C++, Java, Visual Basic, etc. puesto que el proceso de ubicación se basa en un sistema de comparación de arreglos [3], formados por datos adquiridos en tiempo real y datos almacenados en memoria traducidos de los conocimientos previos del terreno.

En la actualidad existen sistemas de reconocimiento avanzados, como el procesamiento de imágenes para el examen de patrones que facilitan y simplifican el trabajo de identificación de ambientes; con la presente investigación se pretendió generar un algoritmo de funcionamiento eficiente que utilizara sistemas embebidos [4] para simplificar el trabajo en hardware requerido para su implementación, así como reducir los procedimientos de procesamiento de señales.

La investigación es realizada por los miembros de los grupos de estudio del Grupo de Investigación en Electrónica de Potencia, Automatización y Robótica (GEPAR), de la Universidad de Antioquia, con el apoyo del estudiante de Maestría Carlos Andrés Madrigal González.

En el presente artículo se presentará de manera detallada y exhaustiva los pasos, procedimientos y secuencias utilizadas para generar un algoritmo que cumpliendo de condiciones iniciales desconocidas cumpliera con las siguientes funciones: 1) calibración de dispositivos de captura de datos para detección de particularidades, 2) recolección de y almacenamiento de puntos singulares para la comparación de estos con las bases de datos disponibles, 3) definición de posición y sentido de la plataforma en el terreno, 4) decidir que es más eficiente: dirigirse hacia un punto conocido del terreno (punto de reporte) o empezar a recorrerlo en busca de un objetivo cuya ubicación es desconocida, las dos sub-acciones anteriormente descritas son subsiguientes e intercambiables, 5) dirigirse hacia un punto de congruencia final donde comunicará las acciones realizadas.

DIGITALIZACIÓN DEL AMBIENTE

Cumpliendo con las restricciones propuestas para el entorno en el cual se ubican las plataformas se debe realizar un proceso exhaustivo de digitalización del mismo, para así disponer de una

base de datos confiable con la cual realizar las comparaciones correspondientes y lograr el objetivo propuesto.

El entorno se describe como una interconexión de trayectorias cerradas, simples y bidireccionales lo cual facilita el proceso de digitalización a través de una matriz multidimensional.

La condición de bifurcación y existencia de trayectorias cerradas del terreno, genera algunas intersecciones de caminos, identificando así, diferentes tipos de datos nodales, *Figura 3*. Cada tipo de nodo se puede identificar con un *label* (normalmente decimal), que sirve para etiquetar, identificar y llevar registro de las singularidades encontradas.

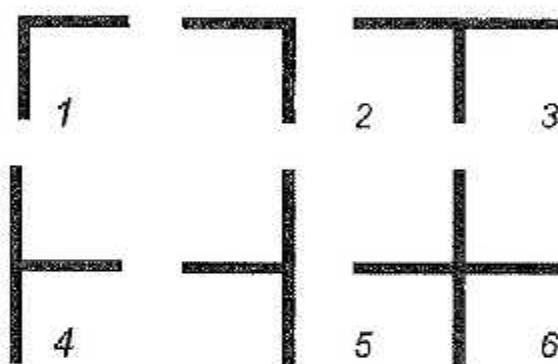


Figura 3, *labels* para la identificación de puntos singulares

Recuerde que aunque se conoce el terreno, la posición de uno de los puntos de reporte y la ubicación de un objetivo final se desconoce la ubicación de uno de los objetivos secundarios y la posición inicial de la plataforma. Es necesario encontrar una trayectoria cerrada que pase por todos y cada uno de los segmentos del terreno que se desean explorar. Para este primer objetivo se puede recurrir a técnicas de inteligencia artificial como las redes neuronales [5] que disponen de facilidades técnicas para la optimización de procesos o si el territorio no presenta gran cantidad de bifurcaciones solo es necesario de cierto tiempo y dedicación para determinar cuál es la secuencia de movimientos que se debe ejecutar para realizar este recorrido *optimo*. Como resultado se obtiene una matriz de m posiciones, row/m ; donde m es el número de segmentos recorridos (note que no es el número de

segmentos del terreno puesto que es posible que se deban repetir trayectorias pero en general se trata de evitar al máximo recorrer dos veces el mismo segmento), donde el dato en la posición i es el tipo de movimiento a realizar en cada uno de los nodos encontrados.

La definición de los tipos de movimientos disponibles es bastante sencilla, se identifican giros de 90° , -90° , 0° y 180° , también se pueden determinar grados de giro intermedios midiendo el tiempo que este demora realizando el desplazamiento, similarmente podemos utilizar *labels* para la identificación de los mismos. Como el terreno es un área de desplazamiento bidireccional es preciso generar a su vez una secuencia de movimientos pero en sentido opuesto al obtenido anteriormente, modificando el vector $rof[m]$ por una matriz bidimensional $rof[2xm]$ donde la primera fila es el recorrido en un sentido y la segunda es el recorrido en sentido opuesto; otra solución al problema del recorrido en sentido opuesto podría ser recorrer el vector original $rof[m]$ en sentido opuesto, ahorrando espacio en el la memoria disponible (normalmente limitada), pero presenta mayores complicaciones puesto que hay que analizar la inversión de tipo de nodos y en general resultaría un procedimiento más tedioso pero igualmente eficiente.

Las posiciones iniciales del vector $rof[2xm]$ no son relevantes porque se trata de una trayectoria cerrada.

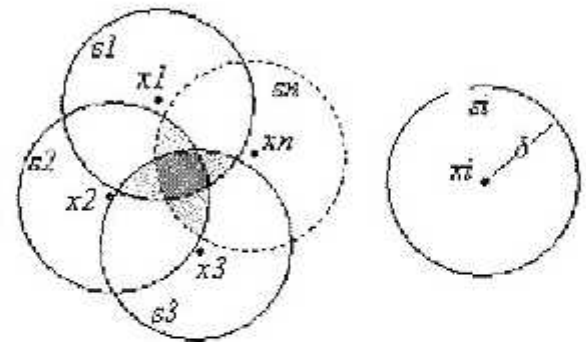
Así se obtiene la digitalización del terreno a través de un recorrido bidireccional que pasa por todos los puntos descados del entorno, esta matriz es utilizada para generar las trayectorias y movimientos que debe realizar la plataforma una vez se ubica en determinado punto del recorrido.

CAPTURA DE DATOS

Puesto que las condiciones iniciales de la plataforma son desconocidas es preciso que se realice un re-calibración de los dispositivos de captura de datos antes de empezar a compararlos. Partiendo de la posición inicial se genera un movimiento aleatorio de tal manera que todos los dispositivos se encuentren aproximadamente en un estado similar definiendo un máximo de

variación con un parámetro; es posible determinar de qué tipo de movimiento se debe realizar haciendo un análisis detallado de las condiciones comunes de todo el terreno, y aunque este desplazamiento se debe generar de manera estocástica, en muchas de las aplicaciones prácticas y académicas, tales como los certámenes mencionados anteriormente el reglamento provee pistas suficientes para la determinación de este accionar inicial.

Para detectar el punto de intersección de los datos de los diferentes sensores se utiliza el concepto matemático de entorno [6], figura 1.



■ : Intersección general

x_n : límite de binarización

s_n : dispositivo de captura de datos

Figura 1. Intersección común de todos los dispositivos de adquisición de datos

Inicialmente el límite de binarización es aleatorio y se realiza el movimiento descrito anteriormente para buscar las posiciones de la plataforma que determinen la existencia de la intersección general.

El entorno nos presenta una serie de valores binarios, pero no se posee información acerca de los datos frontera que definen las condiciones lógicas reales, así estos límites se modifican aplicando proceso anterior definiendo el estado común de intersección, como uno de los estados lógico ya sea "0" o "1" ver, figura 2.

Este proceso se realiza con el objetivo de minimizar los errores en el proceso de adquisición de datos, el cual de es de vital importancia para una correcta comparación la cual evita resultados incongruentes de ubicación.

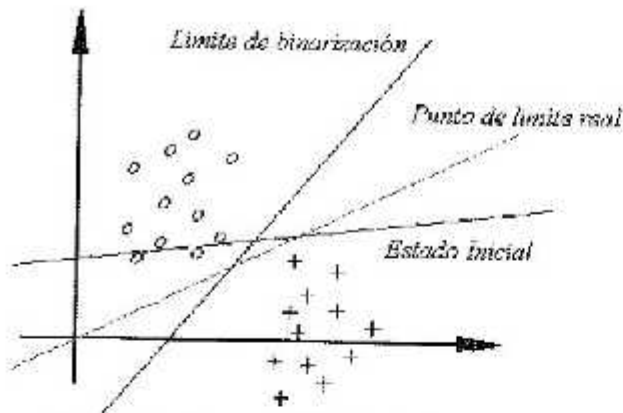


Figura 2. Ajuste del límite de binarización

RECOLECCIÓN DE INFORMACIÓN INICIAL A PARTIR DE PUNTOS SINGULARES

Para un correcto proceso de comparación y ubicación se debe realizar un análisis detallado de las condiciones del terreno, estas deben ser independientes del tiempo y cumplir con las simplificaciones propuestas anteriormente.

La acción subsiguiente es la recolección de un número n de puntos singulares y generar una vector $ub[n]$ donde se guardará el tipo de nodo encontrado en orden cronológico.

El número n de puntos singulares necesarios para determinar con éxito la posición inicial y la posición actual de la plataforma depende y es directamente proporcional a la homogeneidad del terreno de prueba, también varía obedeciendo a las condiciones siguientes a la posición inicial, por esta razón es importante decidir y estandarizar los movimientos que se realizan en cada uno de los nodos mientras la plataforma se encuentra en un punto ciego, que en este caso sería más aproximado hablar de un *segmento de ceguera*, así pues determinamos la manera como la plataforma virará en para cada uno de los nodos identificados, esta asignación se debe ser el resultado de un análisis detallado de los objetivos de la plataforma, se recomienda el movimiento en zig-zag cuando se necesita que esta recorra una mayor parte de terreno aprovechando el intervalo de desconocimiento de la posición, otra alternativa es la generación de trayectorias cerradas durante la ubicación si se busca que se conserve al máximo la posición inicial.

DEFINICIÓN DE POSICIÓN Y SENTIDO

Este es uno de los procesos más exhaustivos puesto que se requiere la generación de una matriz bidimensional [7] que contenga las posibles situaciones encontradas en cada uno de los segmentos del entorno a explorar. El proceso a realizar es simular y obtener para cada uno de los segmentos, las acciones que la plataforma realizaría si este fuese su posición inicial; es necesario aclarar que se debe realizar esta técnica en ambos sentidos de desplazamiento; así pues tenemos una matriz $[2m \times n]$ donde n es el número de nodos necesario para diferenciar individualmente todos los posibles recorridos en el periodo de ceguera, partiendo de las distintas posiciones iniciales, que en este caso serían todos los segmentos del terreno; se define un segmento como: espacio entre dos nodos consecutivos; la experiencia muestra que con el modelo de territorio planteado el parámetro n oscila entre cinco y siete. El dato que se guarda en cada una de las posiciones de la matriz c es el tipo de nodo encontrado; así pues, se tiene una matriz con los datos que la plataforma encuentra si parte de cualquier posición

Se realizan unas modificaciones adicionales a la matriz c , de tal manera que tenga un papel funcional en el algoritmo, agregando dos columnas más, las cuales se conectarán con las matrices ro y ub logrando una ubicación exitosa. Partiendo de cualquier posición inicial pi , y realizando los giros predeterminados para el recorrido ciego en cada uno de los nodos encontrados, después de $n+1$ segmentos la plataforma se encuentra en un segmento pi' , donde ya se tiene el vector ub con la información de los nodos recorridos. La primera columna a generar contiene el sentido en el que la plataforma se encuentra en la posición pi' el cual es un dato binario puesto que consideramos solo caminos bidireccionales, la segunda columna a agregar es la que proporciona una ubicación en el recorrido m , es decir, si se tiene la matriz $ro[2 \times m]$ y definimos la variable $i=0,1,2,\dots,m$, entonces en esta columna adicional se guarda el dato i de la posición que este ocupa en la ruta cerrada ro .

Como los datos de la matriz ro son tipos de movimientos en los nodos y la posición pi' es un segmento no hay mayor complicación en relacionar

pi' , con el nodo inmediatamente anterior o siguiente, para este caso utilizaremos el nodo siguiente.

Ya se tiene los datos necesarios para la ubicación, comparando el vector $ub[n]$ con las primeras columnas de la matriz $c[2mx(n+2)]$, es decir encontrar la fila de la matriz c que cumpla: $ub[0,1,2,\dots,n]=c[j][0,1,2,\dots,n]$, la comparación se realiza con procesos iterativos en el cual se utilizan variables locales para cambiar los datos a comparar [8], llamando j a la variable auxiliar, una vez se logre una comparación exitosa se tendrá un valor único de esta, valor que es de gran ayuda para la continuación exitosa del algoritmo. Cuando existe igualdad entre los vectores anteriores tenemos también gracias a las dos columnas adicionales, el sentido del desplazamiento de la plataforma y el segmento terreno en el que se encuentra, puesto que, como se mencionó anteriormente las posiciones $c[j][n+1]=["1" "0"]$ y $c[j][n+2]=i$, determinan o informan acerca de la posición y el sentido de la plataforma móvil en el recorrido óptimo o matriz $ro[c[j][n+1][c[j][n+2]]$ descrito en los numerales anteriores.

En general la posición actual se resume como:

$$ub[n]=c[j][k] \dots \dots k = 0,1,2,\dots,n$$

$$ro[c[j][n+1][c[j][n+2]]$$

DECISIÓN DE ACCIONES INICIALES

Luego de encontrar la ubicación específica de la plataforma se deben decidir las acciones siguientes que esta realizará, para el caso particular descrito en esta investigación, se poseen dos opciones de comportamiento, estas son necesarias para un correcto cumplimiento de los objetivos, pero no poseen restricciones de orden ni prioridad, así pues, la plataforma puede: primero, dirigirse a un punto de reporte cuyas coordenadas son conocidas (se tiene información acerca del segmento de terreno en el cual se encuentra), ó segundo, iniciar un recorrido minucioso por todo el territorio en busca de un objetivo cuya locación es indeterminada.

Es posible que la plataforma ya se haya encontrado uno de los objetivos, ya sea el punto de reporte o el que tiene ubicación inicial variable y desconocida

durante el recorrido en el cual se encuentra desubicado, o sea, en los segmentos por los cuales esta se desplaza luego de que esta es puesta en el terreno, puesto que el dominio de escogencia de la posición inicial son todos los segmentos del territorio.

Es necesario que aunque el robot este en proceso de ubicación, se lleve un archivo de control de todas las acciones realizadas, así cuando este se ubique satisfactoriamente tenga en su registro datos que influyan en el accionar de este para la optimización del cumplimiento de los objetivos, en este caso, si se realizó o encontró algunos de los objetivos secundarios se continuará con el siguiente y si se cumplieron los dos objetivos, solo restará dirigirse hacia el punto final de reporte cuyas coordenadas son también conocidas.

Tomando inicialmente el caso en el cual no se logró ningún objetivo durante el período de ceguera, para escoger la decisión sobre cual de los dos objetivos pendientes realizar, es preciso utilizar la matriz de digitalización del terreno ro . Como sabemos en que segmento del terreno se encuentra uno de los objetivos, podemos a su vez ubicarlo en una posición l de la matriz ro , así asignamos su posición en el terreno matricializado. Se tiene entonces la ubicación del objetivo l y posición y sentido de la plataforma (las columnas adicionales de la matriz c) y por medio de un proceso simple de seguimiento se puede saber cual es la diferencia numérica de segmentos entre ellos:

$|l - \{c[j][n+1][c[j][n+2] \}$, aportando argumentos contundentes para la acción a realizar, ya sea seguir con el recorrido en la ruta óptima o dirigirse al punto de reporte o primer objetivo.

Cuando se logra cumplir con uno de los objetivos durante el proceso de ubicación, solo es necesario actuar para cumplir el otro: si la plataforma pasó por el punto de reporte cuya ubicación es conocida, hay que seguir recorriendo el territorio según la ruta óptima de la matriz ro ; por el contrario si se detecto el objetivo de posición desconocida se debe realizar los movimientos que la lleven al punto de reporte y finalmente al objetivo final; si se lograron los dos objetivos secundarios, se debe dirigir directamente al objetivo final o meta.

El proceso de desplazamiento el recorrido después de la ubicación se describe a continuación, este proceso utiliza la ayuda de matrices para el control del proceso y la validación de los resultados

DESPLAZAMIENTO Y VALIDACIÓN

Cuando la plataforma ya se encuentra ubicada en el terreno, esta debe empezar a realizar movimientos y acciones que conlleven al cumplimiento de los objetivos; en los numerales anteriores se describieron estos comportamientos iniciales; ahora se desarrollara el tema del desplazamiento y validación de estos movimientos y así poder llevar un control para la prevención y corrección de errores que se pueden cometer en la recolección de datos debido a variaciones inesperadas del terreno.

Para realizar la validación de movimientos y ubicación de la plataforma se debe modificar la matriz ro agregándole una dimensión más. Inicialmente se tiene la matriz $ro(2xm)$ donde cada una de las filas contiene la información acerca de los movimientos que la plataforma debe realizar cuando esta recorriendo la ruta óptima; es decir, la que pasa por todos los segmentos evitando repetir recorridos; en los dos sentidos. La dimensión que se propone agregar tiene la función de proporcionar los datos de validación de los nodos encontrados, para realizar el seguimiento de la trayectoria y corregir inmediatamente errores de ubicación.

Adicionando una matriz con el mismo número de columnas y filas; $2xm$; cuyo espectro de datos son los tipos de nodos (datos decimales definidos anteriormente) que la plataforma detectara mientras se desplaza durante el recorrido de la ruta óptima si su ubicación física coincide con su ubicación virtual. La continuación de datos descrita anteriormente proporciona la ayuda necesaria para poder realizar el proceso de validación de desplazamiento puesto que en todo momento después de la ubicación si se esta recorriendo la ruta óptima, se sabe que nodo se encontrará y que moviendo se debe realizar. El proceso es exhaustivo puesto que necesita que para cada recorrido de ruta óptima se guarden los datos del nodo encontrado y el giro a realizar pero es eficiente en su

funcionamiento y no presenta errores de incongruencia en su algoritmo.

El proceso de captura de datos sobre el nodo encontrado y tipo de giro a realizar puede ser aplicado a su vez en desplazamientos hacia cualquier dirección fija puesto que en ocasiones es necesario que la plataforma ya ubicada se dirija a un punto específico cuyas coordenadas conocemos, el desarrollo a seguir para este problema es paralelo al descrito, se deben generar rutas rápidas en los dos sentidos, desde del punto de partida (cualquier segmento del terreno), hasta el punto final tomando datos de los nodos encontrados y el tipo de giro a realizar, teniendo así un registro de direccionamiento desde cualquier posición inicial hasta un segmento final determinado, puesto que los recorridos pueden tener diferentes longitudes, es necesario agregar una bandera de activación para indicar que el recorrido se completo, normalmente una solución puede ser agregando una nueva posición a cada uno de los recorridos el cual debe ser único e identificable para la detección del fin del camino.

Cuando el nodo encontrado no coincide con la clase que se encuentra en la tercera dimensión de la matriz ro se concluye que la plataforma cometió un error en su ubicación y nuevamente se encuentra pérdida indicando que el proceso de locación se debe reiniciar

CONCLUSIÓN

El algoritmo descrito en este artículo presenta gran cantidad de restricciones de operabilidad, no indicando deficiencia en campos de acción. Es necesario que se entienda el proceso como una salida alternativa a la solución de un problema común en actividades académicas de competencias robóticas relacionadas con la ejecución de actividades en laberintos guiados por una línea negra y la relación estrecha que existe en la utilización de matrices multidimensionales y la solución de problemas de ubicación. El algoritmo no presentó problemas de operación, pero su aplicación requiere una gran dedicación por parte de los interesados cuando se demande hacer exámenes minuciosos de las condiciones del

terreno para interpretar el funcionamiento de la plataforma según todas las condiciones que se puedan presentar durante el transcurso del recorrido, además es necesario una larga etapa de prueba-error, puesto que, este como modelo teórico se tiene que modificar y adaptar a las posibles limitaciones que le impongan los dispositivos de captura de datos, los cuales según se observó representa la mayor fuente de errores en el proceso de ubicación.

Se observó que los procesos de ubicación, validación, desplazamiento y demás son eficientes tanto en tiempo como en funcionamiento pero se tuvo el inconveniente del el espacio en memoria que debe estar disponible debido a la gran cantidad de información almacenada como matrices, tanto constantes como variables.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Betancur Betancur, Manuel. (). *Introducción a al Mecatrónica*. Medellín. Editorial Universidad Pontificia Bolivariana.
- [2] Pagina de A + D -Olimpiada
- [3] Grossman, Stanley L., (1996). *Algebra Lineal*. Quinta Edición. Mexico D.F. McGraw Hill
- [4] *Sistemas Embebidos*
- [5] Del Brío, Bonifacio Martín. Sanz Molina, Alfredo, (2005). *Redes Neuronales y Sistemas Difusos. Segunda Edición Ampliada y Revisada*. MADRID, España. AlfaOmega Ra-Ma.
- [6] Leithold, Louis. (1998). *El Cálculo*. San Antonio. Oxford University Press
- [7] Bronson, Gary J., (1999). *C++ para Ingenierías Ciencias*. Mexico. Thompson Editores.
- [8] Oviedo Regino, Efraín, (2005). *Lógica de Programación*. Bogotá. Ecoe Ediciones

AUTORES

CESAR AUGUSTO URIBE MENESES, Estudiante de Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad de Antioquia, miembro de los grupos de estudio del grupo de investigación GEPAR, intereses: Redes Neuronales e Inteligencia Artificial.
 ecaum388@udea.edu.co
 (4) 4126180- (316) 5350451

JAIME IGNACIO MARIN VARGAS, Estudiante de Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad de Antioquia, miembro de los grupos de estudio del grupo de investigación GEPAR, intereses: Robótica, Automatización, Animación 3D, Procesamiento digital de Imágenes e Inteligencia Artificial.
 jaimemari17@gmail.com
 (4) 4962105- (310) 3954706

LEONARDO ARANGO BAENA, Estudiante de Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad de Antioquia, miembro de los grupos de estudio del grupo de investigación GEPAR, intereses: Animación 3D, Robótica, Redes neuronales y Procesamiento digital de imágenes.
 larango_baena@hotmail.com
 (4) 4122124- (300) 3042790

ANDRES FELIPE PEDRAZA MONSALVE, Estudiante de Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad de Antioquia, miembro de los grupos de estudio del grupo de investigación GEPAR, intereses: Redes Neuronales, Automatización y robótica.
 grantombobadil@hotmail.com
 (4) 2673706 (312) 8055015