

# Obra de Arte Programable

INTRODUCCIÓN

DE LA INGENIERÍA  
AL ARTE

MÁQUINA AUTOMÁTICA  
PROGRAMABLE

HAROLD COHEN

CÓDIGO

ROOM FULL OF  
EMPTINESS

MATRIZ

TRABAJO  
A FUTURO

CONCLUSIONES

FUENTES  
DE CONSULTA

CAMPOS POTENCIALES  
ARTIFICIALES





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



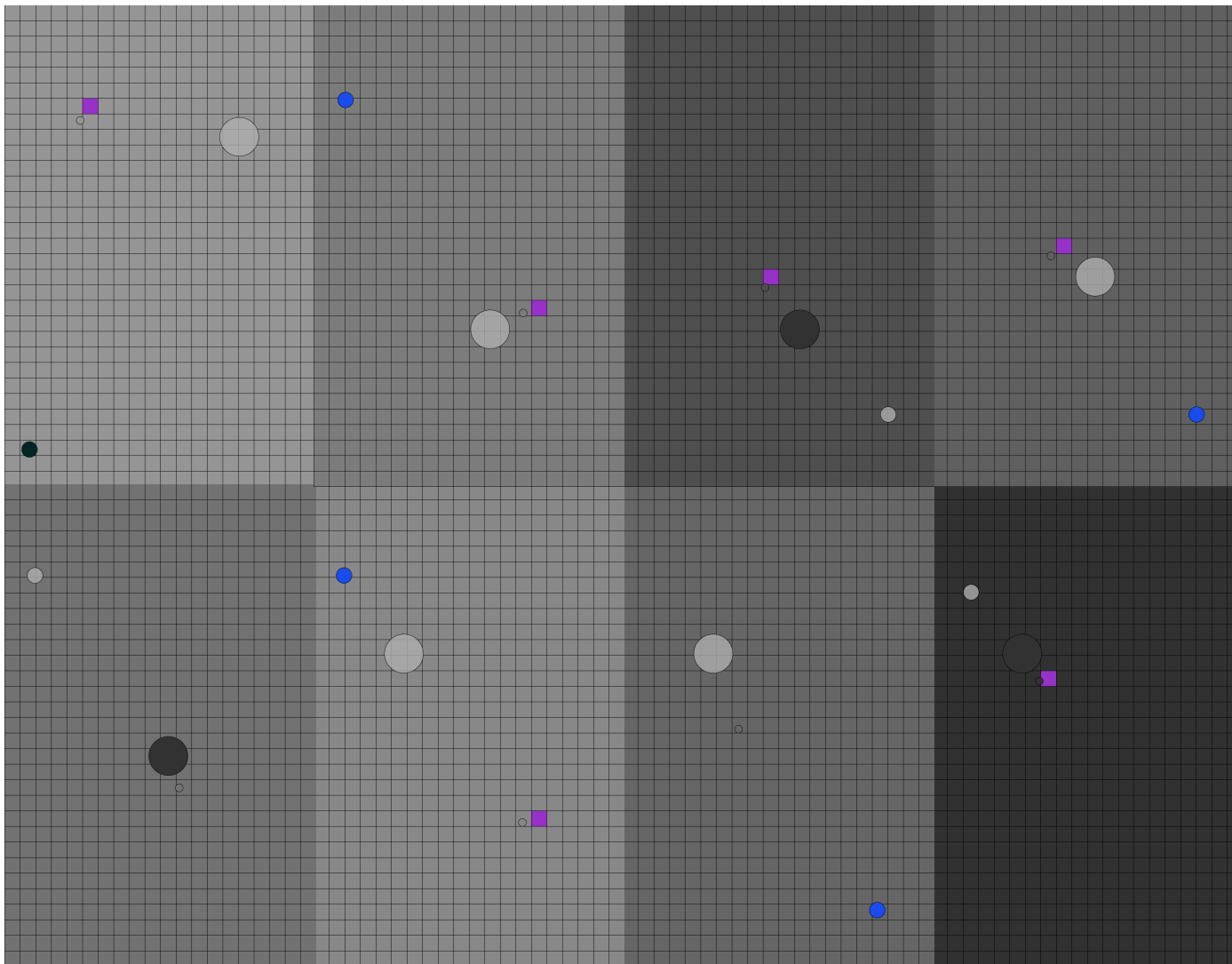
**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## CAMPOS POTENCIALES ARTIFICIALES



*Ilustración 46- Secuencia de imágenes del campo potencial programado en Processing. El círculo grande representa el obstáculo, el mediano el objeto de atracción y el mas pequeño, siempre acompañado de un cuadro, el robot o l objeto que es atraído, el cuadro va marcando los cuadros que el robot va pisando.*

Durante la investigación se trabajó con el modelo de Campos Potenciales Artificiales y se propuso en un inicio como modelo de interactividad para el proyecto RoFuE. Finalmente se descartó por la *Red Virtual* que en esencia representara más claramente el concepto de la *visión sesgada del mundo*.

Presento los campos potenciales artificiales aquí como parte del proceso de trabajo en el que entendí, que es en el modelo matemático y el código de programación donde radica uno de los núcleos más importantes de las obras de arte electrónico.

Entender cómo funciona nos da la posibilidad de modificar la forma en que reacciona, esto significa básicamente modificar su discurso.

Un artista electrónico al poseer conocimiento en ambas áreas, arte-ciencia, puede ver las relaciones entre ambas. Percibiendo los atributos expresivos en el desarrollo de un modelo matemático que se use para la realización de una obra, así como visualizar la interactividad de un sistema y pensar en cómo implementarla mediante algún modelo que sea útil.

Los campos potenciales artificiales son un modelo matemático que se utiliza en la robótica para la cinemática inversa de brazos manipuladores, como los que hay en las fábricas. Así como en la navegación de robots móviles que se desplazan mediante un sistema de ruedas o piernas (Khatib, 1986).

El modelo matemático de los campos potenciales me parece poético, visto como una representación matemática del deseo y la atracción. El destino ejerce una fuerza que atrae ineludiblemente al robot hacia él, pero en el camino hay obstáculos que dependiendo sus características provocarán que el robot los rodee, atravesase o simplemente lo paralicen en un mínimo local<sup>40</sup> del que el robot no podrá salir. Todos los elementos en ese espacio componen el campo potencial, modifican con cada movimiento las fuerzas que están en juego. La danza de fuerzas contenida en el campo potencial traza la ruta que seguirá el robot para alcanzar su destino.

Este modelo se pensó como una estrategia para hacer más eficiente y robustos (en ingeniería se utiliza el término robustez para decir que un sistema es estable ante los cambios) el desplazamiento de los robots que no se encuentran en un ambiente controlado, sobre entendiendo que el robot necesita alcanzar su destino de forma eficiente sin chocar con ningún obstáculo en su trayecto, pero ¿qué pasaría si modificáramos el modelo del campo potencial para que no trazara rutas eficientes?

Por ejemplo mostrando un comportamiento errático, temeroso ante el destino, inmune a los obstáculos, o inclusive inmune a la atracción del destino. Esto podría convertir al robot en un laboratorio de comportamientos. Las reacciones y rutas trazadas pueden relacionarse de forma metafórica a un reflejo completamente vivo y humano, por ejemplo el de la atracción que genera un deseo de ser consumado (Ilustración 47).

<sup>40</sup> El mínimo local es una situación que aparece cuando el robot no puede calcular la ruta debido a la forma del obstáculo.



El entender el modelo abre la posibilidad de cuestionarlo con situaciones hipotéticas<sup>41</sup> y permite modificarlo para hacer que un robot pueda reaccionar de formas poco eficientes pero discursivamente potentes representando así una idea (incluso desarrollándola), en vez de ser solo un sistema de desplazamiento.

A continuación se presenta un extracto del código programado en Processing, que muestra una de las clases principales del código, la clase Cell, que es donde se llevan a cabo la mayor parte de los cálculos del campo potencial y donde se concentran los datos de la posición de cada uno de los elementos presentes. El texto que va entre “/\* ..... \*/ “ y el que sigue después de // o más, son comentarios dentro del programa que explican a que responden los valores de las variables, las constantes y las funciones dentro del programa, también, encabezando el código esta la descripción del mismo y los autores.

La intención de mostrar este fragmento del código (la clase Cell) es presentar esa capa oculta donde descansa la interactividad que puede ofrecer este modelo matemático. Esta Clase contiene todas fórmulas del modelo matemático (Véase EL MODELO MATEMÁTICO DE LOS CAMPOS POTENCIALES ARTIFICIALES pag. 123) y se ejecutan en cada movimiento que haga el robot.

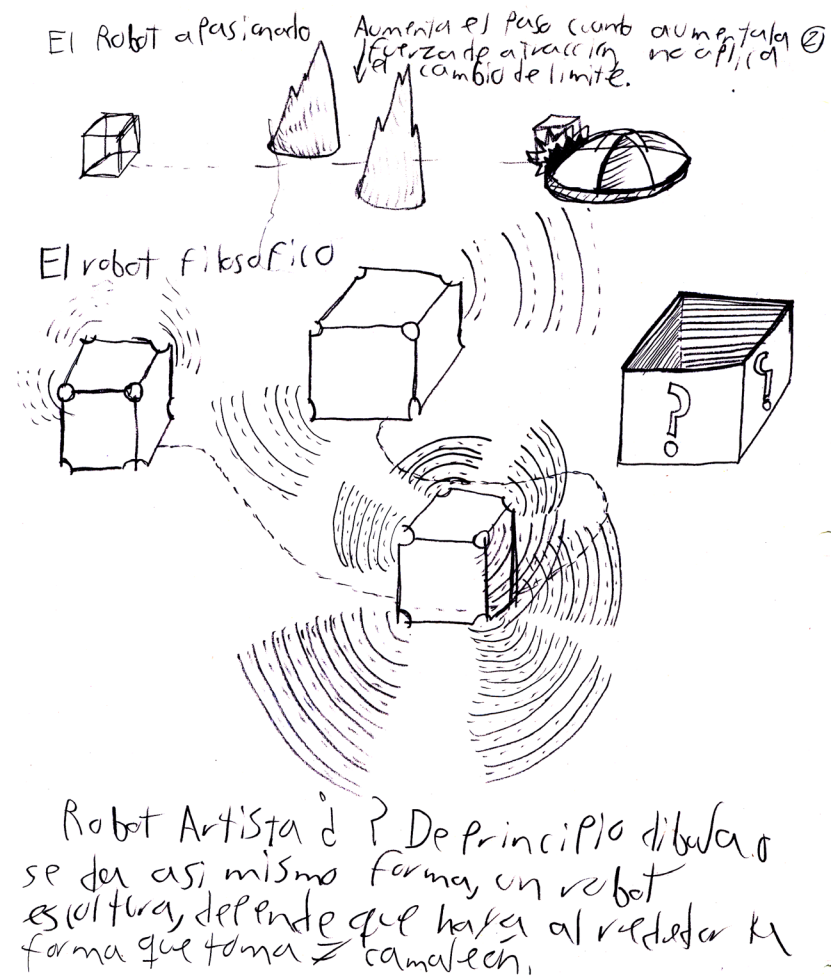


Ilustración 47- Bocetos de los campos potenciales pensados como un laboratorio de comportamientos. Aquí el robot se representa como un laboratorio de comportamientos en el que lo que importa es denotar conductas de un ente atraído sobre su objeto de atracción a diferencia de la habitual visión en la que el robot debe realizar su trabajo de la forma más eficiente posible.

<sup>41</sup> Me refiero a cuestionarlo desde un punto de vista artístico, es decir viendo las capacidades expresivas y discursivas que tiene y como se modifican al cambiar la forma en la que reaccionaria el robot.

```

/*
 *CAMPOS POTENCIALES
 *Por: Joaku de sotavento, Baruch y Ernesto Huerta
 *Los campos potenciales son un modelo matemático
 *utilizado en la planeación de trayectorias para sistemas
 *dinámicos.
 *Aquí se plantean los campos potenciales para el uso en sistemas
 *de representación expresiva para fines artísticos
 *A continuación se presenta la Clase Cell que es donde se calculan
 *las fuerzas que dan movimiento a los elementos dentro del campo
 *potencial.
 */

```

```

class Cell {
float x, y;
////////// componest of attractor field
float fatx;
float faty;
////////// componets of resulsive field
float d0; // distance between the obstacle's and the robot
float rep; // the vector between the robot and the obstacle
float frepx; // repulsion force on x
float frepy; // repulsion force on y
//////////
float fTotalx;
float fTotaly;
color col;

```

```

/// constructor of grid
Cell(float ex, float why, float fTotx, float fToty) {
x = ex * _cellSize;
y = why * _cellSize;
fTotalx = fTotx;
fTotaly = fToty;
col = 100;
}
////////// Este método calcula la fuerza de atracción entre el
robot
//y su destino dependiendo de la razón de cambio Delta I (interés
del //robot)
PVector attraccion(Robot r, Attractor att) {
float d = dist(r.loc.x, r.loc.y, att.loc.x, att.loc.y);
//println(d);
if (d <= r.deI) {
fatx = (att.E) * (r.loc.x - att.loc.x);
faty = (att.E) * (r.loc.y - att.loc.y);
}
else {
fatx = (att.E) * (r.loc.x - att.loc.x);
faty = (att.E) * (r.loc.y - att.loc.y);
}
return new PVector(fatx, faty);
}

```



```

PVector repulsion(Robot r, Obstacle[] obs) {
for ( int i = 0; i < obs.length; i++) {
d0 = dist(r.loc.x, r.loc.y, obs[i].loc.x, obs[i].loc.y);
if ( d0 <= obs[i].rangoO) {
rep = ((-1*obs[i].N) * (1/d0 - 1/obs[i].rangoO)) * (pow(((pow((r.
loc.x - obs[i].loc.x), 2)) + pow((r.loc.y - obs[i].loc.y), 2)), 3/2));
    }
else {
rep = 0;
    }
frepx = rep * (r.loc.x - obs[i].loc.x);
freply = rep * (r.loc.y - obs[i].loc.y);
    }
return new PVector (frepx, frepy);
    }
//////***** Este es el vector unitario de la fuerza de repulsión y
////la fuerza de atracción
PVector fuerzaT () {
fTotalx = (fatx + frepx) / sqrt(pow(fatx, 2)+ pow(frepx, 2));
fTotaly = (faty + frepy) / sqrt(pow(faty, 2)+ pow(frepy, 2));
return new PVector (fTotalx, fTotaly);
    }

```

```

void drawMe(Robot rob, Attractor att) {
float colorCamb = dist(rob.loc.x, rob.loc.y, x, y);
stroke(0);
float colorCamb2 = dist(rob.loc.x, rob.loc.y, att.loc.x, att.loc.y);
fill(map(colorCamb2, 0, dist(0, 0, width, height), 0, 255));
if (colorCamb < _cellSize/2) fill(150, 50, 200);
rect(x, y, _cellSize, _cellSize);
    }

```

Ir a:



ROOM FULL OF EMPTYNESS



EL MODELO MATEMÁTICO



PROPUESTA DE IMPLEMENTACIÓN DE LOS CAMPOS



POTENCIALES ARTIFICIALES



ACERCA DE COMO EL MEDIO ES EL TEMA



INTERFAZ



VISIÓN COMPUTACIONAL

## EL MODELO MATEMÁTICO

Este sistema comprende la operación de robots en tiempo real en ambientes complejos donde las condiciones del entorno no son controladas y pueden estar cambiando constantemente. El modelo de los campos potenciales artificiales depende de una constante retro alimentación a través de los sensores del robot que le digan en qué punto del mapa se encuentra y si hay obstáculos cerca, a partir de estos datos el robot calcula su trayectoria paso a paso esquivando los obstáculos que va percibiendo en su ruta.

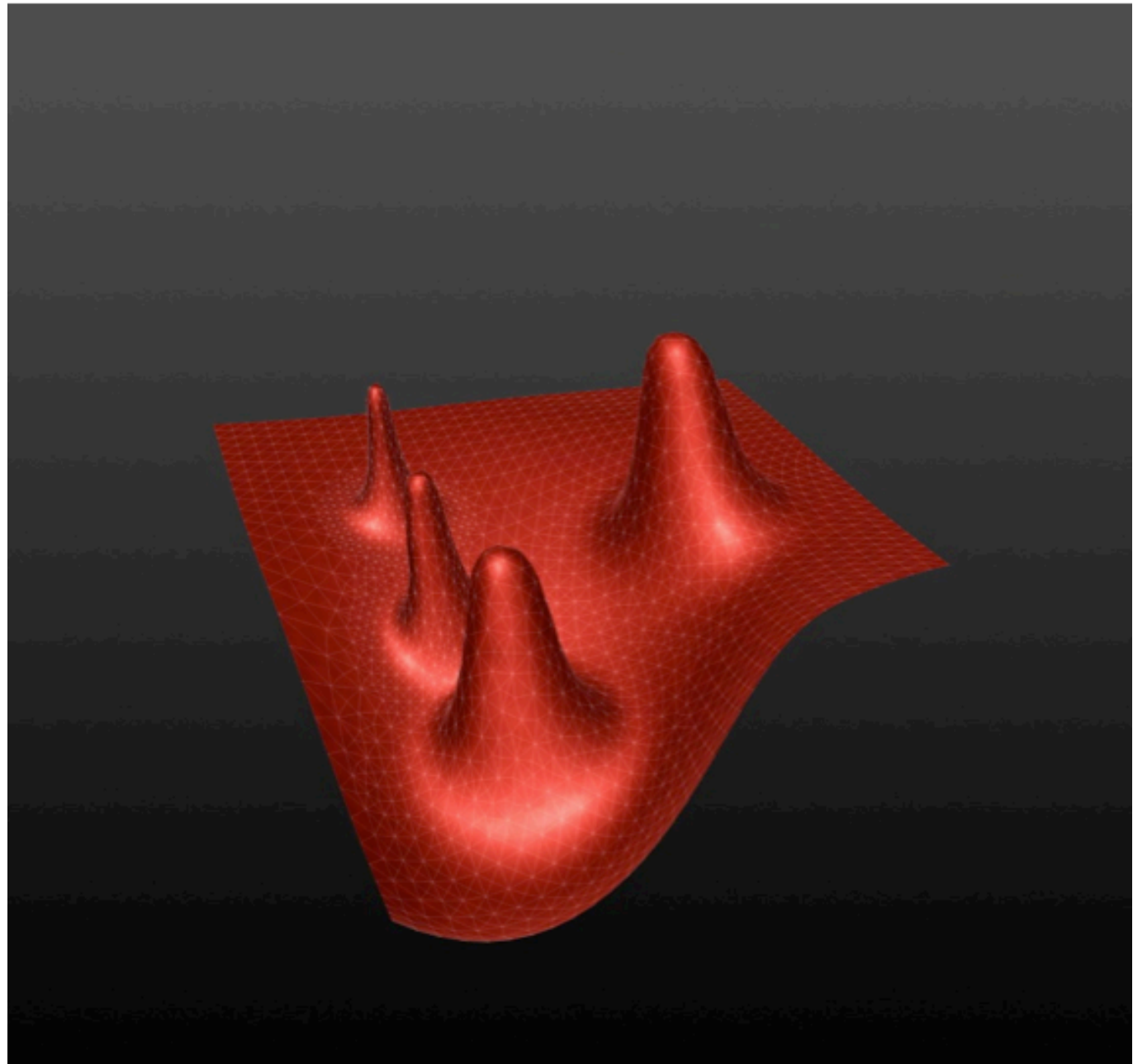
Por ejemplo en un brazo manipulador, que fue en el primer caso en el que se aplico este enfoque de planeación de trayectorias, se plantea “...como un control directo sobre el movimiento del manipulador en el espacio operacional que es el lugar donde se describe originalmente la tarea, en contraste con un control de acciones correspondiente al espacio de movimiento de las articulaciones que se obtienen a partir de la transformaciones cinemáticas” (Khatib, 1986). Es decir se concibe la planeación de los movimientos del robot tomando en cuenta la relación espacial del entorno con él, a diferencia del otro enfoque en el que se hace esta planeación pensando en los movimientos angulares que debe hacer el robot para alcanzar una posición final, es un cambio de concepción del espacio que concibe en un caso al robot como centro y en el otro se percibe al robot como otro elemento del ambiente.

Este enfoque en el artículo de Khatib aunque concibe su aplicación para robots móviles se enfoca en brazos manipuladores (Khatib, 1986). Para un enfoque en robots móviles citare a Arambula Cosío, F. y Padilla Castañeda, M. A. quienes lo definen de la siguiente manera:

“La idea central es generar fuerzas de atracción y repulsión en el entorno de trabajo del robot para guiarlo hacia su objetivo. El objetivo ejerce una fuerza de atracción en el robot y cada obstáculo tiende a alejar al robot de si para evitar una colisión”(Arambula Cosío & Padilla Castañeda, 2003).







*Ilustración 48- Representación 3d de un campo potencial, donde el valle representa el punto de mayor atracción y las cimas los campos de repulsión.*

El campo potencial se modela como una licra tensada de forma horizontal y es jalada en algún punto hacia abajo, este representa el destino. Hay protuberancias en la licra que representan los obstáculos, entre mayor sea la protuberancia mayor es su campo de repulsión. Entonces el robot es representado como una bola de billar que se suelta sobre la licra e ira librando los obstáculos hasta llegar a la parte más baja.

A continuación se expone el modelo de campos potenciales que posteriormente fue programado en el lenguaje de programación de Processing.

$$\bar{F}(\bar{q}) = F_{attr}(\bar{q}) + \sum_{i=1}^N F_{rep_i}(\bar{q}) \quad (1)$$

En esta fórmula (Eq. 1) se explica cómo: las fuerzas que se ejercen sobre el robot  $F(q)$  son la suma de la fuerza de atracción  $F_{atr}$  mas la fuerza de repulsión  $F_{rep}$  de todos los obstáculos en el campo.

Para modelar esta malla de fuerzas se toma en cuenta al menos un robot  $q$  sobre el que se ejerce una fuerza de atracción por parte del destino  $q_{dest}$  y uno o varios obstáculos  $q_{obst}$  que ejercen fuerzas de repulsión que alejan al robot de estos.

La fuerza de atracción se obtiene multiplicando una constante ajustable de atracción, del destino,  $E$  que multiplica a la distancia entre el robot y su destino (Eq. 2). Como la fuerza de atracción puede llegar a ser muy grande dependiendo de la distancia, y esto

podría ocasionar que la velocidad del robot fuera tal que atravesara los obstáculos, ya que la fuerza de atracción sería superior que la de repulsión, o inclusive que se estrellara con el destino, se aplica una condición en la que si la distancia es mayor que un parámetro establecido  $d_{ref}$  la fuerza de atracción es la constante de atracción multiplicada por la normal de la distancia entre el robot y el destino, es decir se multiplica por uno (Eq. 3).

$$F_{attr}(\bar{q}) = \begin{cases} E(\bar{q} - \bar{q}_{dest}) & |\bar{q} - \bar{q}_{dest}| \leq d_{ref} \quad (2) \\ E \frac{(\bar{q} - \bar{q}_{dest})}{|\bar{q} - \bar{q}_{dest}|} & |\bar{q} - \bar{q}_{dest}| > d_{ref} \quad (3) \end{cases}$$

También se puede entender de forma sintética a la fuerza de atracción  $F_{atr}$  (Eq. 4 y 5) como:

$$F_{attr}(\bar{q}) = E\hat{q}_a \quad (4)$$

Donde

$$\hat{q}_a = \frac{(\bar{q} - \bar{q}_{dest})}{|\bar{q} - \bar{q}_{dest}|} \quad (5)$$

La fuerza de repulsión (Eq. 6) se calcula por cada uno de los obstáculos  $q_{obs}$  solo si el rango de repulsión del obstáculo está en la trayectoria del robot. Esta fuerza se calcula multiplicando la constante ajustable de la repulsión del obstáculo  $-N$  multiplicado por 1 que es dividido entre la magnitud del robot y el obstáculo, menos 1 entre la amplitud del campo de repulsión, todo esto multiplicado por la distancia entre el robot y el obstáculo, dividido entre la magnitud al cubo del robot y el obstáculo. Si el obstáculo no está en la trayectoria del robot la fuerza de repulsión es 0.

$$F_{rep}(\bar{q}) = \begin{cases} -N \left( \frac{1}{|\bar{q} - \bar{q}_{obs}|} - \frac{1}{do} \right) \frac{\bar{q} - \bar{q}_{obs}}{|\bar{q} - \bar{q}_{obs}|^3} & \text{si } |\bar{q} - \bar{q}_{obs}| \leq do \\ 0 & \text{cc.} \end{cases} \quad (6)$$

Una vez calculadas la fuerza de atracción y la de repulsión se obtiene el vector unitario<sup>42</sup> (Eq. 7) que es utilizado para ir calculando las coordenadas a las que se desplazara el robot a cada movimiento.

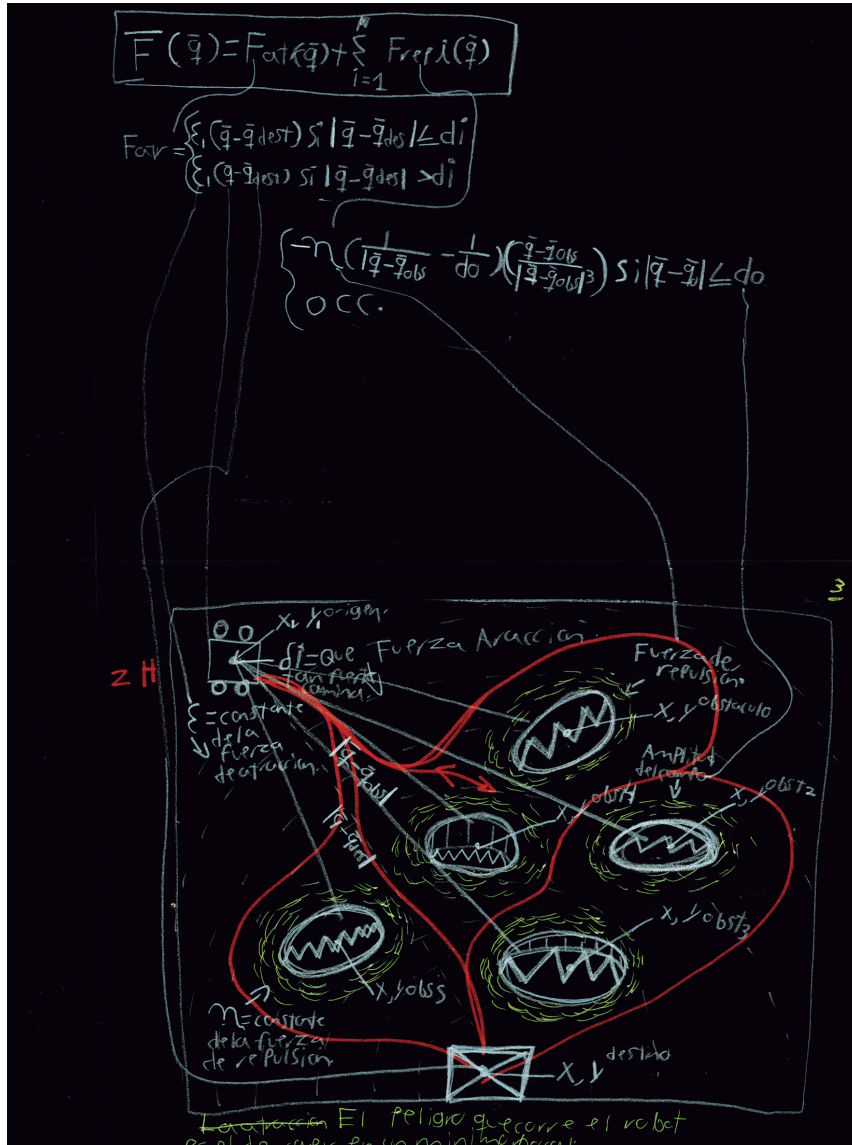
$$f(\bar{q}i) = \frac{F_{atr} + F_{rep}}{\sqrt{F_{atr}^2 + F_{rep}^2}} \quad (7)$$

La siguiente formula (Eq. 8) es utilizada para calcular los pasos del robot,  $qi+1$  es la posición siguiente del robot que es igual a la posición del robot  $qi$  menos  $di$  que es el impulso del robot multiplicado por el vector unitario  $f(qi)$  el cual está formado por las fuerzas de atracción y repulsión en el campo.

$$\bar{q}i+1 = \bar{q}i - di * f(\bar{q}i) \quad (8)$$

Estos cálculos se hacen para cada uno de los movimientos del robot, se pueden hacer cientos de veces por segundo, dependiendo de la velocidad del procesador, de la capacidad de los sensores y de la cantidad de obstáculos en el campo. Por eso es posible que el robot vaya adaptándose a un lugar con condiciones cambiantes y así re-planear su ruta.

<sup>42</sup> El vector unitario es un vector cuya longitud es igual a 1. Este se obtiene de dividir el vector entre su longitud (Poole, David., 2011).



Ir a:





-  CAMPOS POTENCIALES ARTIFICIALES
-  PROPUESTA DE IMPLEMENTACIÓN DE LOS CAMPOS POTENCIALES ARTIFICIALES
-  MODULO DE VISION DE RoFuE
-  K-MEDIAS [K- MEANS]

Ilustración 49. Boceto de los Campos Potenciales, con las fórmulas que hacen funcionar cada una de las partes del campo potencial y señalando que datos provee cada actor del campo potencial. Las líneas que trazan curvas entre los campos de repulsión son las posibles rutas que tomaría el robot para alcanzar su destino.

## PROPUESTA DE IMPLEMENTACIÓN DE LOS CAMPOS POTENCIALES ARTIFICIALES

Se pensó la posibilidad de adaptar los campos potenciales artificiales como sistema de interacción de la instalación, asignando aleatoriamente a cada uno de los usuarios que entren uno de los elementos del modelo, robots, obstáculos o destinos.

Se vincularía la activación de los sonidos, el paneo y volumen a las variables que actualiza el sistema en tiempo real: distancia entre el robot y el destino, distancia entre el robot y el obstáculo, la fuerza de atracción actual, la fuerza de repulsión y la posición del robot. Vinculando estas variables con las luces y sonido de la instalación se podría haber generado una interacción en tiempo real, generando una constante retroalimentación del usuario con la instalación y viceversa.

Este modelo de interacción crea un ambiente reactivo el cual es muy diferente a un ambiente determinista que sin embargo es por el que se optó al final.

Un modelo determinista, según la definición de Stuart Russell (Russell & Norving, 2009) es:

*...un modelo en el cual el siguiente estado (es decir la siguiente acción del sistema) está completamente determinada por el estado actual y la acción que está ejecutando en este caso el robot.*

Un modelo determinista dependiendo de la complejidad de sus respuestas y la multiplicidad de combinaciones que sea capaz de ofrecer puede crear la ilusión de diálogo entre el usuario y la obra, sin embargo esta interacción está predefinida de antemano y todas

las opciones han sido revisadas y programadas sus respuestas.

Los campos potenciales por el contrario no tienen respuestas pre-programadas, modelan el espacio en tiempo real y calculan el sistema reactivo de fuerzas con los elementos que son percibidos, determinando un robot, obstáculos y un destino, no prevé las respuestas sino que analiza a cada instante el espacio, las relaciones de atracción y repulsión que se generan en el campo y en base a eso modifica las variables del sistema.

El modelo reactivo responde tomando en cuenta las condiciones actuales del ambiente y sus personajes, y el determinista activa una respuesta pre-programada que corresponde a una posibilidad de estar en ese lugar, en ese momento y de esa forma lo cual genera la ilusión de una respuesta natural, pero es solo ilusoria.

El modelo reactivo presenta el atractivo de la respuesta inmediata. En la instalación RoFuE se proponía que relacionara los sonidos de los estratos sonoros con las distancias entre el usuario y el destino cambiando su paneo dependiendo su ubicación en relación al centro de la habitación, y su volumen a razón de su cercanía con el destino.

El modelo reactivo es más parecido a la forma en la que circula la información en nuestra época donde tenemos una percepción simultánea de los sucesos, por ejemplo en el terremoto del 11 de marzo del 2011 en Tokio en el que en menos de 10 minutos de haber sucedido, había videos en youtube (el portal de video en Internet de

nuestra época) desde el punto de vista de personas que vivieron ese suceso. Tenemos una sensación de simultaneidad en la que gracias a los medios de comunicación y los diferentes dispositivos como celulares inteligentes tenemos una imagen de como se sincronizan sucesos de diferentes partes del mundo en el mismo momento.

Myron Krueger (Krueger, 1977) apuesta a que la obra de arte (hablando por supuesto del arte electrónico) no está en ninguno de los elementos que componen la obra y por supuesto no está basada en la *contemplación* de imágenes, sonidos o formas sino en la manera en la que el usuario se relacionan con la obra, es decir en la *interactividad* de la pieza, Krueger se plantea esto mientras realizaba la parte tecnológica de una instalación artística llamada *Glow Flow*, él explica:

*“Mientras Glow Flow fue visualmente muy exitosa, se logró más bien como una escultura cinética que como un entorno responsivo. Sin embargo, Glow Flow me dejó ante una serie de decisiones:*

*1 -. El arte interactivo es un rico medio de composición muy distinto de lo que concierne a la escultura, las artes gráficas o la música.*

*2 -. Para responder de manera inteligente, la computadora debe percibir lo más posible sobre el comportamiento de los participantes.*

*3 -. Con el fin de concentrarse sobre la relación entre el medio ambiente y los participantes en lugar de solo enfocarse en*

*los participantes, sólo un pequeño número de personas debe estar involucrado al mismo tiempo.*

*4 -. Los participantes deben ser conscientes de cómo el ambiente está respondiendo a ellos.*

*5 -. La elección del sonido y el sistema de respuesta visual debe ser decidido por la capacidad de transmitir una amplia variedad de relaciones conceptuales.*

*6 -. Las respuestas visuales no deben ser juzgadas como arte ni el sonido como música. La única preocupación estética es la calidad de la interacción.*

Tomando como referencia a Krueger el sistema determinista puede generar en la Instalación de RoFuE un comportamiento en el que el usuario establece más una relación de contemplación, ya que él realiza su trayecto, el sistema lo analiza y compone para el usuario una respuesta puede apreciar y tratar de relacionar con su recorrido. En el caso del sistema reactivo, se establece una relación donde la experiencia más fuerte no es la de la contemplación sino la interacción, donde lo que pasa está sucediendo todo el tiempo y no hay que detenerse a observar cómo reacciona la instalación, ya que todo el tiempo está reaccionando.

El problema con los campos potenciales, aunque responden al ideal de Krueger sobre lo que ha de ser un sistema interactivo, es más actual y se adapta más a las formas de circulación de la información, se habría sacrificado el concepto central de la pieza, el

de la mirada sesgada, que se presenta de forma más narrativa en ese dialogo de transitar y esperar la respuesta del sistema.

El modelo de interactividad basado en los campos potenciales se puede pensar como una IUI [Interfase de Usuario Intangible], parafraseando el modelos de TUI [Interfase de Usuario Tangible] propuesto por el Media Lab de MIT (Ishii, Lakatos, Bonanni, & Labrune, 2012). En las interfaces tangibles, a diferencia de las interfaces gráficas, se interactúa con objetos reales que sirven de vínculo o mando dentro del mundo virtual. En las interfaces intangibles se interactúa con el cuerpo como vehículo presionando botones virtuales que no ves, dándole mayor peso a elementos como el sonido que igualmente es intangible.

Ir a:

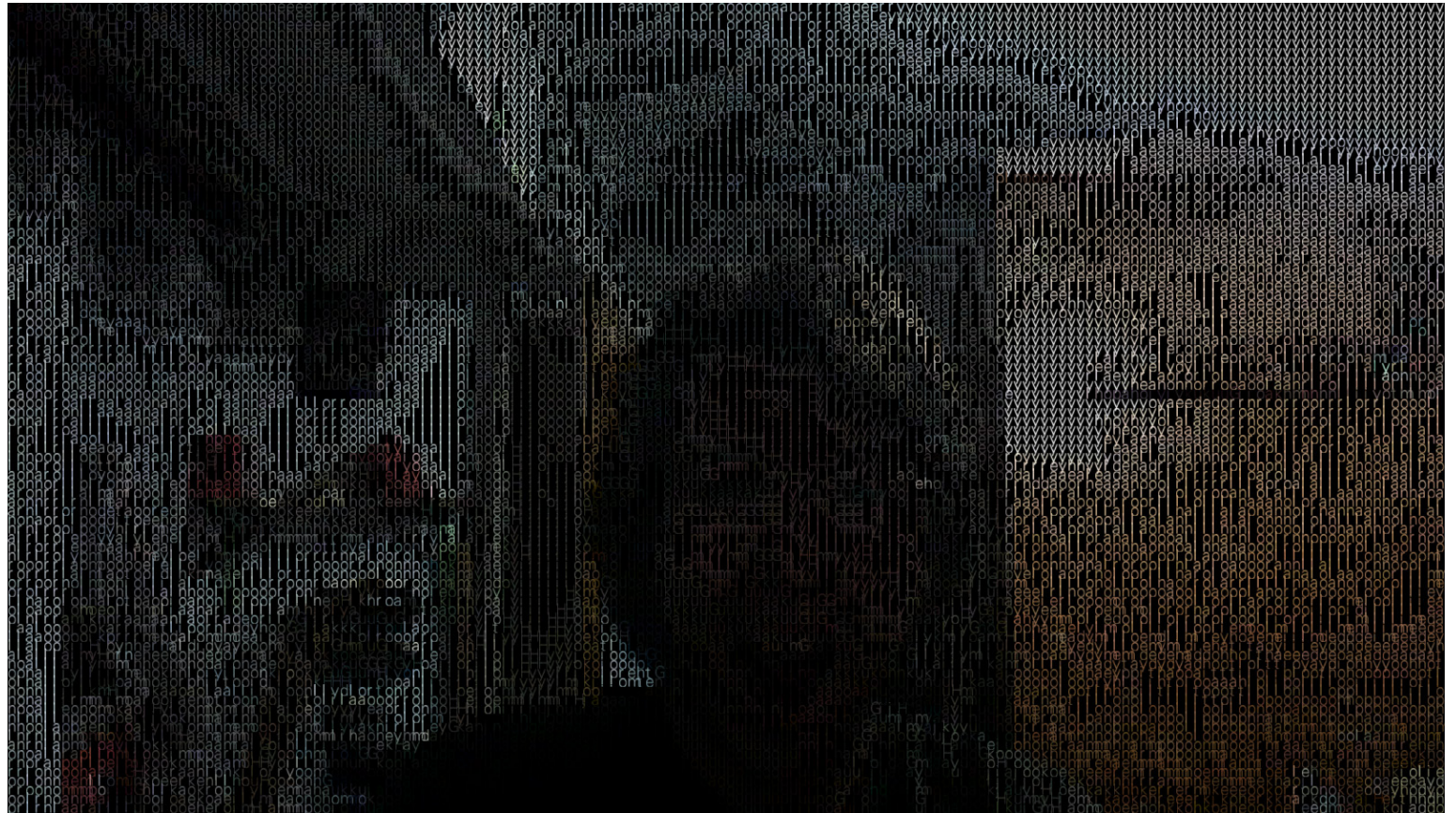


CAMPOS POTENCIALES ARTIFICIALES	EL SEL
EL MODELO MATEMÁTICO	
MODULO DE VISION DE RoFuE	
CÓDIGO	
MÁQUINA AUTOMATICA PROGRAMABLE	
CONCLUSIONES	





## MODULO DE VISION DE RoFuE



*Ilustración 50- Captura de imagen del código ASCII Video de Ben Fry, tomado de la biblioteca de video para Processing GSVideo. 2013.GSVideo. 2013.*



Se optó por un sistema de visión<sup>43</sup> en vez de una matriz de sensores porque los sistemas de visión tienen la ventaja de poder trabajar con casi toda la información que entre en el campo visual de la cámara. En vez de ubicar y calibrar 300 sensores, solo se hace para 2 cámaras, lo cual además hace más fácil detectar fallos en el sistema, ya que el cableado se reduce dramáticamente, permitiendo localizar el problema en hardware de manera más eficiente.

Las cámaras que se usarán para el rastreo en RoFuE son cámaras RGB-D (del tipo Kinect). La cámara Kinect entrega información de RGB y de profundidad. Con esa información mediante la librería de OpenNI y SimpleOpenNI podemos visualizar en *Processing* el *mapa de profundidad*, la *nube de puntos* rastrear a una persona frente a la cámara, entregar su silueta recortada y localizar 24 puntos de su anatomía con sus coordenadas X, Y, Z (entre ellos cabeza, cuello, torso, cintura, y hombros, codos, manos, muñecas, cuello, puntas de los dedos, cadera, rodillas, tobillos y pies, tanto izquierdo como derecho)(Borestein, 2012).

Con toda esta información se puede hacer lo que sea (Ilustración 51), aplicarles un algoritmo, utilizarlos para ponerles un sistema de partículas cambiarlos de color, moverlos al tocar un punto virtual etc.

La forma que se elijo para hacer el rastreo de la persona es por medio de la nube de puntos o point cloud<sup>44</sup> combinada con

<sup>43</sup> Para ahondar en los que es la visión computacional Véase VISIÓN COMPUTACIONAL pag. 161

<sup>44</sup> En el portal de Point Cloud Library (PCL, 2013) se describe la nube de puntos de la siguiente forma: "La nube de puntos (point cloud) es una estructura de datos usada para representar una colección multidimensional de puntos y es comúnmente usada para representar información 3D. En una nube de puntos 3D los puntos suelen representar sus coordenadas geométricas X, Y y Z resaltando las superficies muestreadas. Cuando hay información de color presente la nube de puntos se vuelve 4D.

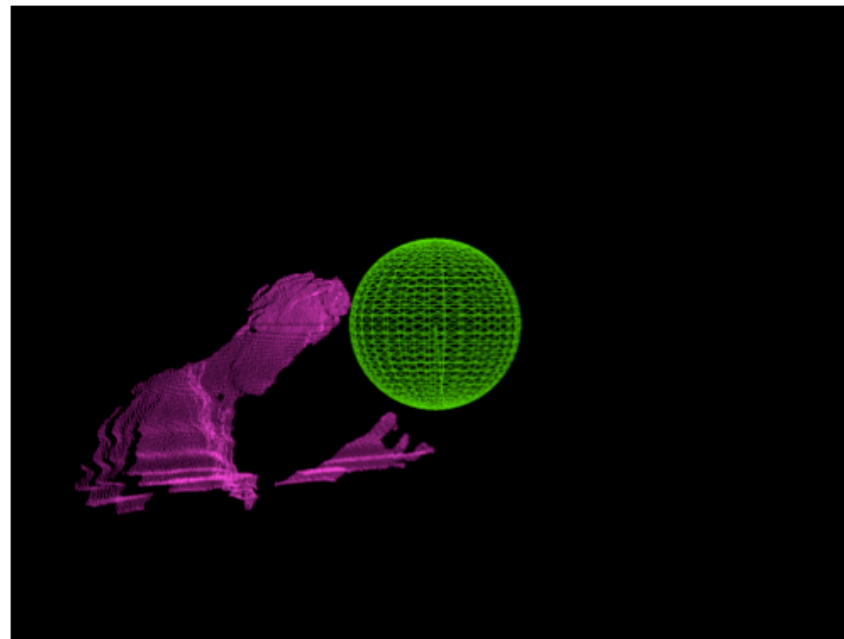


Ilustración 51. Captura, nube de puntos con ruido de perlin en Processing. 2013.2013.

un procesamiento K-Medias (K-Means) y un rastreador Kalman. Por lo regular en muchos proyectos de Arte Electrónico se adaptan bibliotecas ya programadas a las necesidades del proyecto lo cual es lo más práctico<sup>45</sup>, y habría sido maravilloso poder hacerlo así, de hecho se utilizaron las bibliotecas de OpenNI que nos sirven para usar todas las funciones ya descritas de la Kinect, se planeaba utilizar la librería de OpenCV para el filtro de K-Medias y Kalman. Las bibliotecas de OpenNI y OpenCV no están especialmente diseñadas para proyectos artísticos, más bien son utilizadas en entornos científicos e industriales. Los lenguajes más usados para

<sup>45</sup> Esto del código se ponía más y más intrincado, casi siempre está toda la información en Internet y si estas librerías hubieran estado completas solamente se habría tenido que ir haciendo las llamadas a las librerías para incluirlas en el código, que podría haber salido mal.

programar con estas bibliotecas son C, C#, C++, Java y Python por mencionar algunos.

El lenguaje de programación que he estado usando hasta ahora para el proyecto es un lenguaje basado en Java llamado Processing, que hasta ese punto de la investigación había sido suficiente. En Processing junto con la versión de C utilizada para Arduino programe todos los proyectos que desarrolle en la maestría. Para implementar el sistema de rastreo en RoFuE hay un inconveniente: los algoritmos de Kalman y K-medias de la biblioteca de OpenCV no están disponibles para Processing. Este lenguaje es muy popular entre estudiantes de ingeniería, diseñadores y artistas electrónicos, está diseñado a un nivel más alto y no es tan poderoso como otros lenguajes. Esto es para hacerlo accesible a públicos menos especializados y por lo regular depende de bibliotecas que hacen de intermediario como JavaCVPro y SimpleOpenNI que sirven para utilizar en Processing OpenCV y OpenNI.

Después de instalar estas bibliotecas exitosamente logré obtener la nube de puntos y programar un filtro que dividiera por colores los objetos que se encontraran a diferentes distancias (Ilustración 52). Hacía falta implementar K-medias para identificar cada cumulo de puntos como sujetos y Kalman para el seguimiento, pero JavaCVPro no tiene implementado ninguno de los dos. Estuve buscando alguna biblioteca para Processing que tuviera estas funciones programadas pero no la encontré. Lo más que fueron códigos que visualizaban en abstracto los algoritmos, pero ninguno estaba implementado para su aplicación en visión.

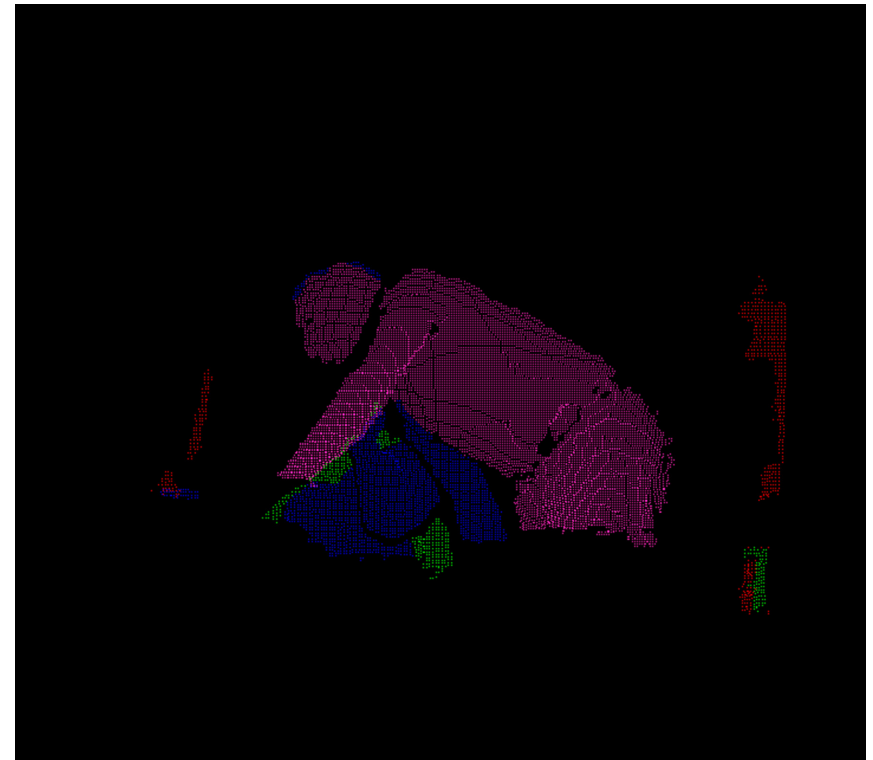


Ilustración 52- Captura, nube de puntos con un filtro de distancias, programado en Processing, 2013.

La opción era programarlos yo mismo con ayuda de los expertos de visión del laboratorio de Biorobotica del Posgrado de Ingeniería de la UNAM. Surgió otro problema, ellos programan en C++, donde las bibliotecas de OpenCV se pueden usar plenamente. Esta era la segunda vez que buscaba una biblioteca para Processing que no estaba implementada, la primera vez fue con los campos potenciales artificiales.

La oportunidad perfecta había llegado, tenía que programar un filtro desde cero.

Ir a:



ROOM FULL of EMPTYNESS

PROPUESTA DE IMPLEMENTACIÓN DE LOS CAMPOS POTENCIALES ARTIFICIALES

K-MEDIAS [K- MEANS]

ACERCA DE COMO EL MEDIO ES EL TEMA

VISIÓN COMPUTACIONAL



## K-MEDIAS [K- MEANS]

El closterizado de *k-Medias* (k-Means) es un método de análisis de clusters<sup>46</sup> que tiene como objetivo particionar  $n$  observaciones en  $k$  clusters, en la que cada observación pertenece al cluster con la distancia más cercana, esto da lugar a una partición del espacio en células de Voronoi (Kanungo, Mount, Netanyahu, Piatko, Silverman, & Wu, 2002).

k-Medias es un sistema basado en los diagramas de Voronoi (Ilustración 53). Estos dividen un espacio en grupos definidos por una serie de semillas repartidas en el plano. Se calcula la distancia euclidiana de cada punto en el plano a la semilla y así se busca cual de las semillas es la que está más cerca, asignando ese punto a esa región. Esto logra una división del plano donde cada parte de él corresponde a una de las semillas (Miu, 2001).

k-Medias es un método estocástico que no garantiza encontrar la mejor solución pero encuentra de forma rápida una solución que sea suficiente.

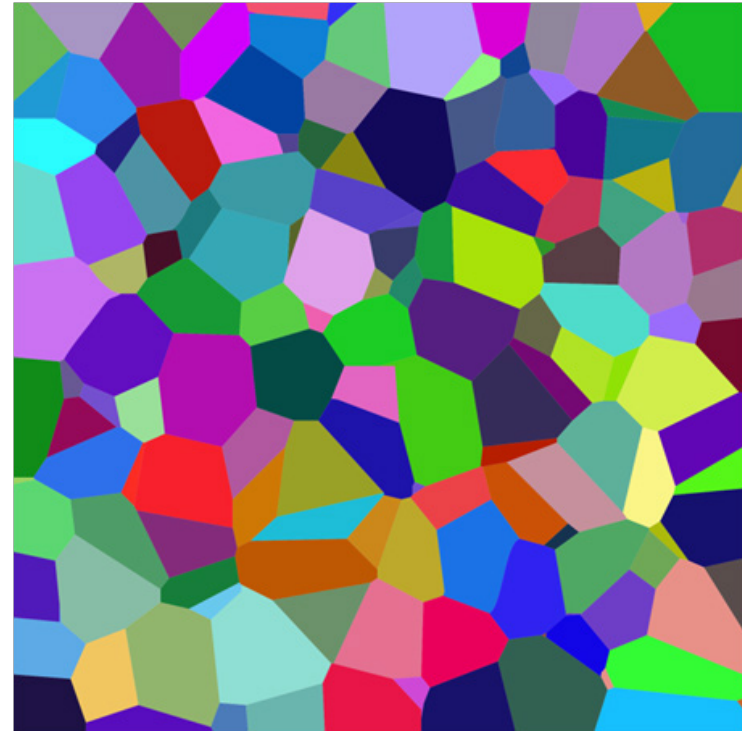


Ilustración 53. Capa a color de una diagrama de Voronoi 3D. Imagen tomada de Wikipedia.

<sup>46</sup> Una clusterización es una segmentación de una población heterogénea en cierto número de subgrupos más homogéneos o ubicar ciertos grupos en un conjunto de datos basados en ciertos criterios de similitud (Estivill-Castro).

A continuación se expone el modelo de k-Medias y como se visualiza cada parte programada en Processing.

Dado un cumulo de puntos que pueden ser: los pixeles de una imagen, puntos en el espacio, localizaciones geográficas, personas en un lugar... (Eq. 9).

$$D = (x_1, \dots, x_n) \quad x_i \in \mathbb{R}^d \quad (\text{Eq. 9})$$

Donde  $D$  es un conjunto de datos que pertenecen a un espacio dimensional  $\mathbb{R}^d$  (Ilustración 54).



Ilustración 54. Primer parte del algoritmo de K-medias en la que se arrojan puntos a lo largo y ancho del lienzo de forma aleatoria.

Se arrojan aleatoriamente *centroides*  $M$  a lo largo y ancho de la imagen. El numero *centroides* se seleccionan de ante mano, de forma empírica, buscando el mejor resultado mediante prueba y error (Eq. 10).

$$M_1, \dots, M_k \in \mathbb{R}^d \quad (\text{Eq. 10})$$

Donde  $M$  son los centroides que se esparcen aleatoriamente en el espacio dimensional  $\mathbb{R}^d$  (Ilustración 55).

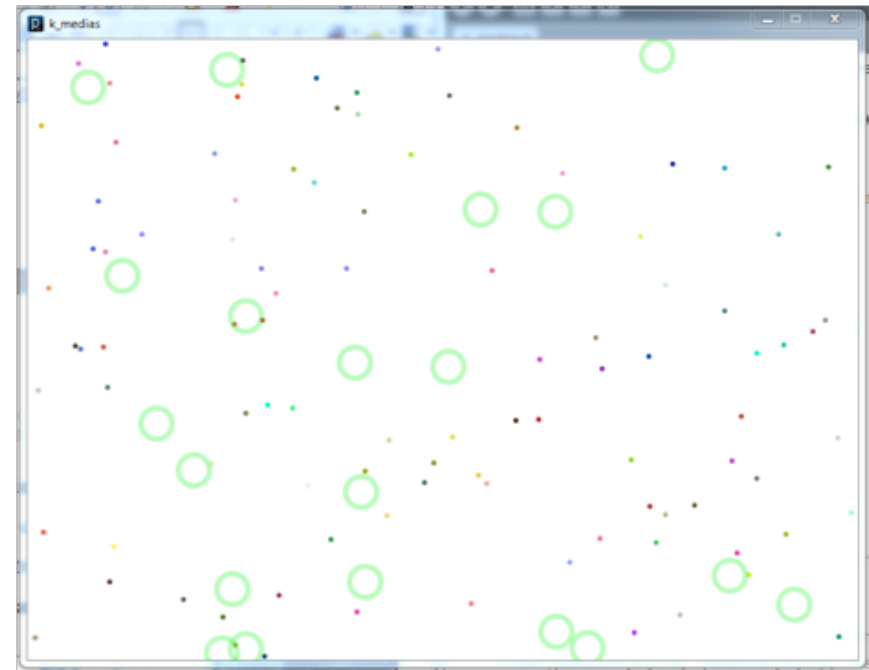


Ilustración 55. Segunda parte del algoritmo de K-medias. Sobre los puntos en el lienzo se arrojan centroides de forma aleatoria, el número de centroides es decidido arbitrariamente.



Se calcula la distancia entre cada uno de los puntos  $a_{ij}$  a todos los centroides  $M$  arrojados (Ilustración 56), poniendo un índice a cada punto que indica que centroide está más cerca (Eq. 11).

$$\sum_{j=1}^k \left( \sum_{i=1}^n a_{ij} \|x_i - m_j\|^2 \right) \text{ donde } a_{ij} = \begin{cases} 1 & \text{si es asignada sobre } a_{ij} \\ 0 & \text{cc.} \end{cases} \text{ (Eq. 11)}$$

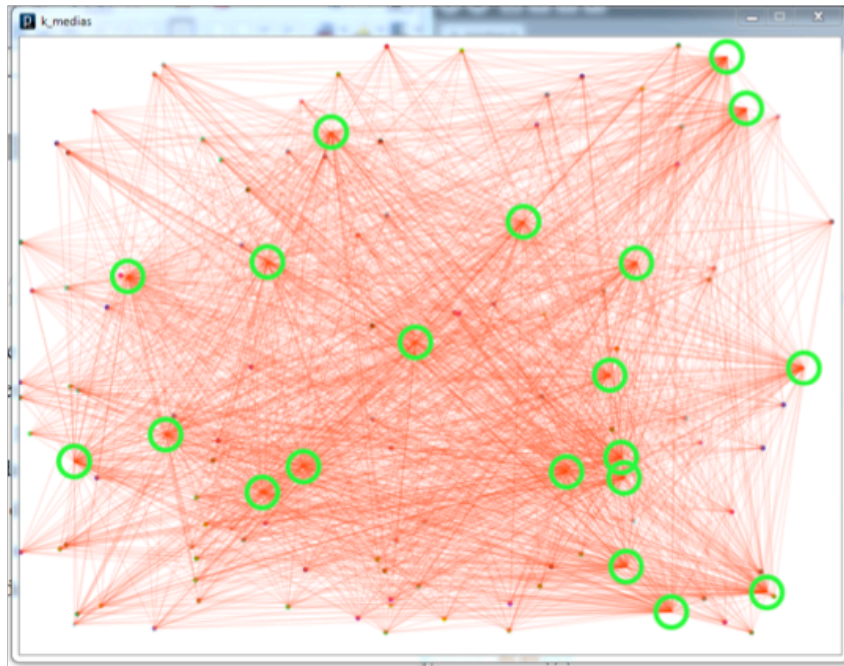


Ilustración 56. Tercera parte del algoritmo de K-medias. Se obtiene la distancia de cada punto con todos los centroides asignándole el índice del centroide que tiene más cerca.

Se obtiene la media entre todos los puntos que se definieron cercanos a ese centroide para obtener las coordenadas del centroide de ese grupo.

Luego se reubican los centroides, eliminan los que están en el mismo sitio para reducir su número y de esta forma se forman los grupos o clusters, coloreando de un color diferente a la sección que correspondiente a cada centroide en particular (Ilustración 57).

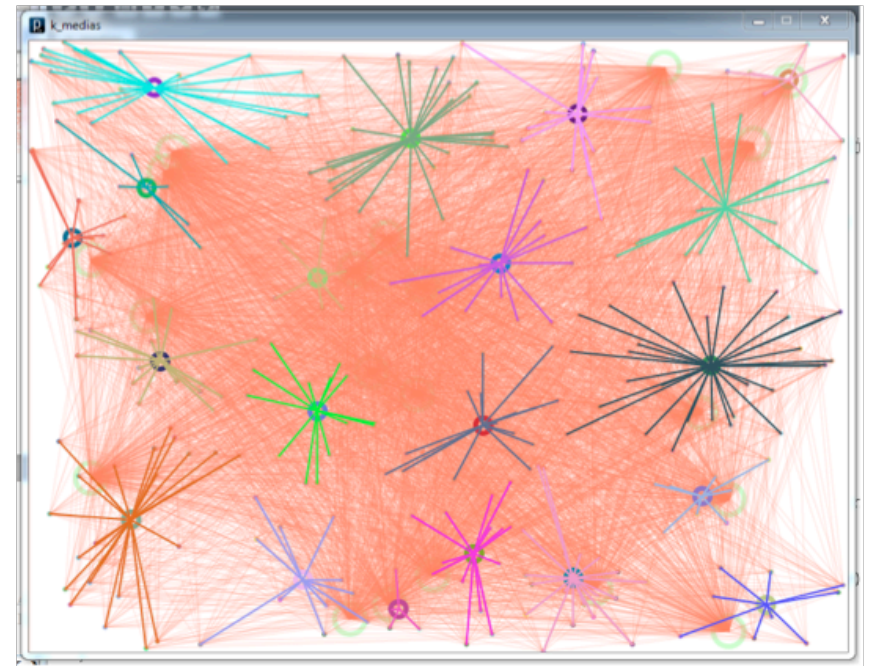


Ilustración 57- Cuarta parte del algoritmo. Se saca la distancia media de todos los puntos que coinciden en su índice con el mismo centroide para obtener las coordenadas del centroide de ese grupo y se reubica el centroide en su nueva posición.

En RoFuE estos clusters después de pasar por un filtrado de k-Meadias, dividirán a la nube de puntos de cada persona en la habitación.

Una vez que k-Meadias identifique los centroides de cada nube de puntos, se necesitara usar un filtro de Kalman para que no se pierda o confunda el centroide obtenido y así poder rastrear a la persona. Las coordenadas obtenidas de cada una de las personas y su altura activarán las células virtuales de la retícula y marcará el estrato sonoro en el que se encuentra.

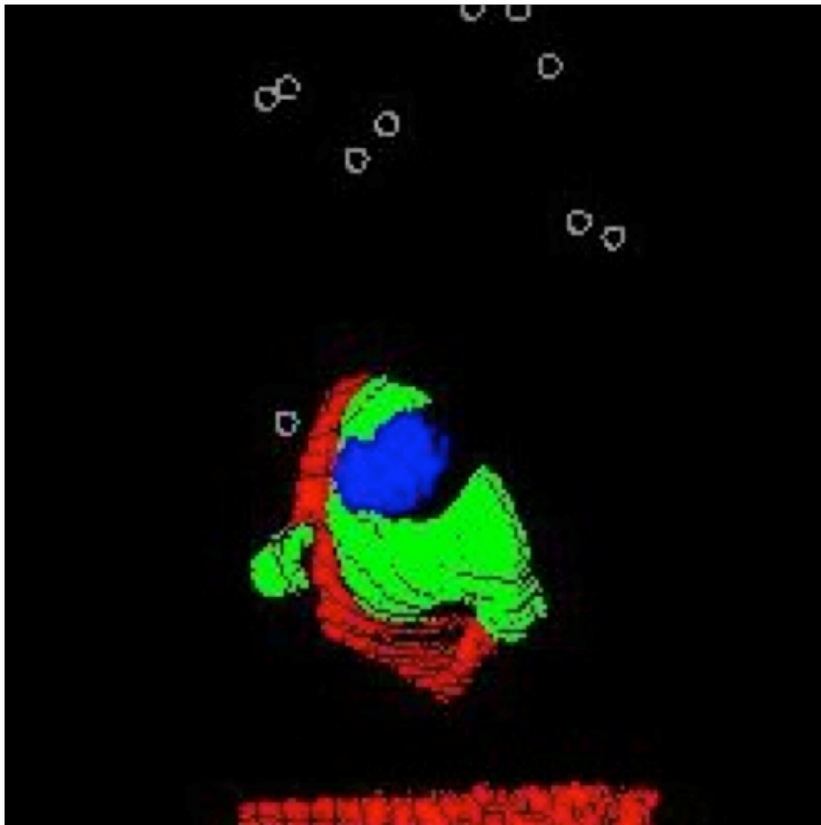


Ilustración 58. Aquí se muestra como se reparten los centroides aleatoriamente sobre la nube de puntos generada por la Kinect.

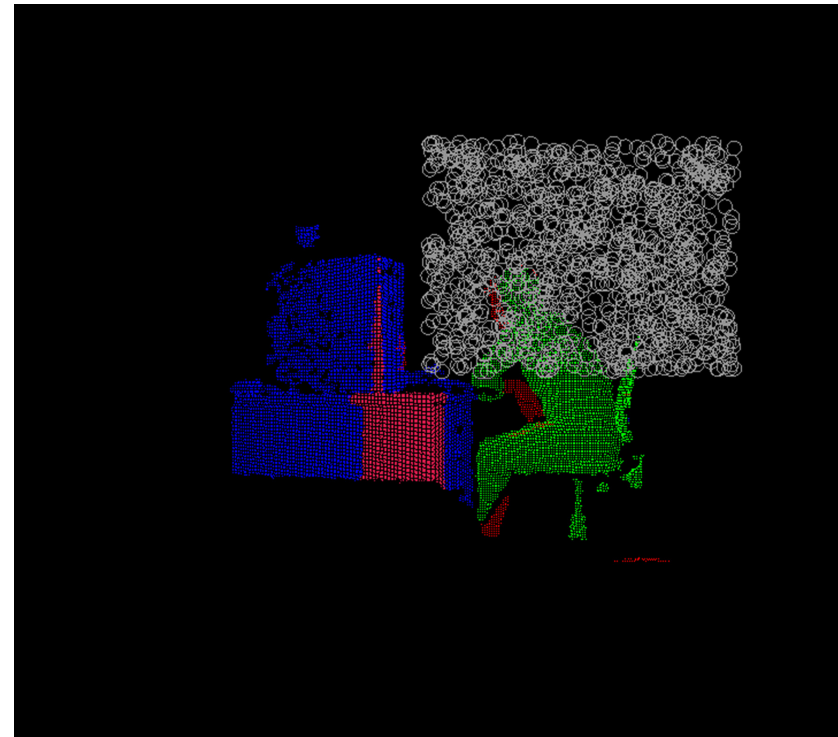


Ilustración 59. Nube de puntos en vista aérea, con filtrado de distancias a colores. Pensado para obtener la altura del usuario y los centroides del algoritmo de K-medias.

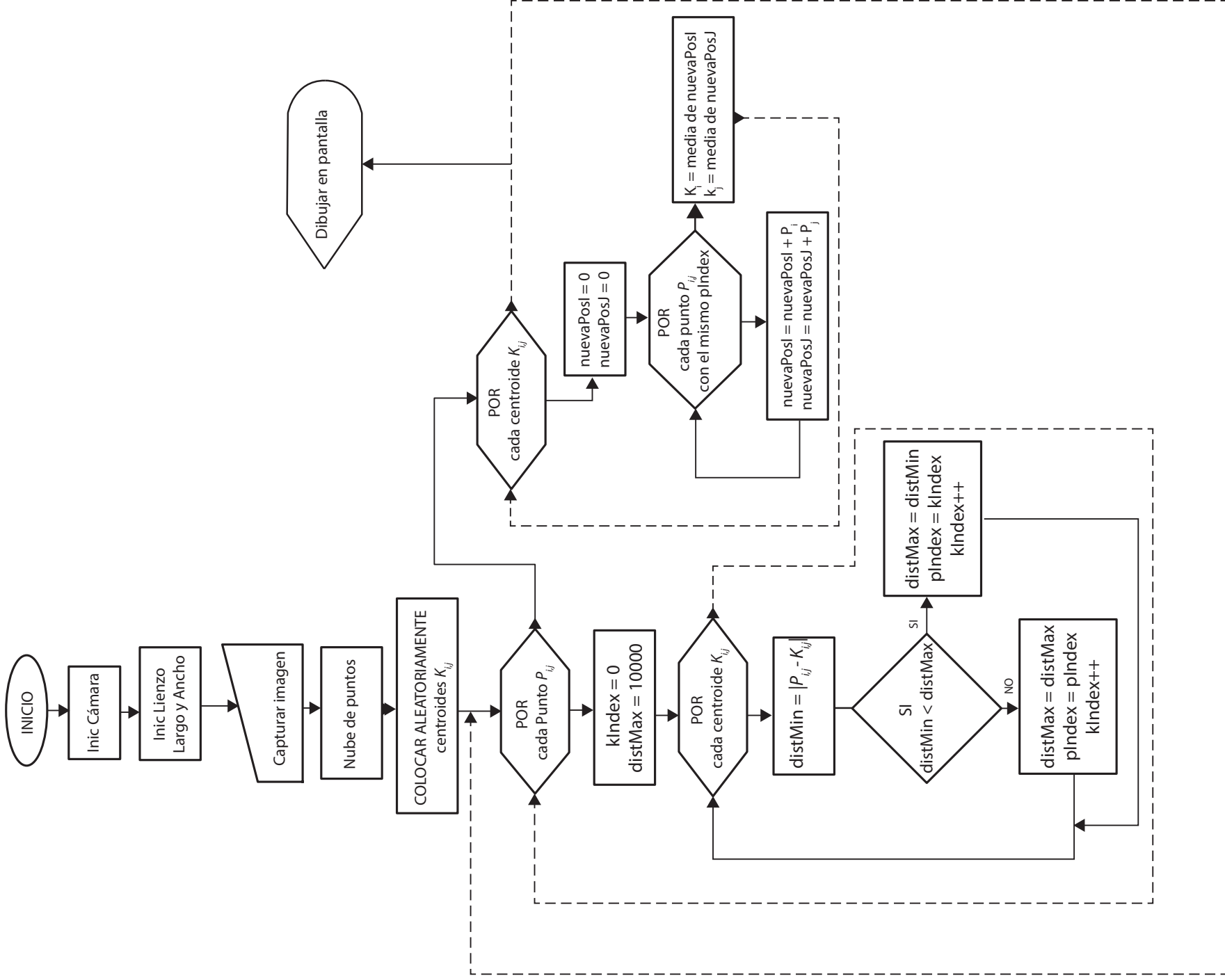


Ilustración 60- Representación gráfica del algoritmo de K-medias.





INICIO

Capturar imagen

Obtener Nube de puntos

Colocar aleatoriamente los centroides

```
POR (p = 0; p < pTotal; p++)
  kIndex = 0
  distMax = 10000
```

```
POR (k = 0; k < kTotal; k++)
  distMinima =  $|P_{i,j} - K_{i,j}|$ 
```

```
SI (distMinima < distMaxima)
```

```
  distMax = distMin
```

```
  pIndex = kIndex
```

```
  kIndex++
```

```
SI NO
```

```
  distMax = distMax
```

```
  pIndex = pIndex
```

```
  kIndex++
```

```
POR (k = 0; k < kTotal; k++)
```

```
  nuevaPosI = 0
```

```
  nuevaPosJ = 0
```

```
POR (p = 0; p < pTotal; p++)
```

```
   $K_i = \text{nuevaPosI} + P_i$ 
```

```
   $K_j = \text{nuevaPosJ} + P_j$ 
```

```
   $K_i = \text{media de nuevaPosI}$ 
```

```
   $K_j = \text{media de nuevaPosJ}$ 
```

DIBUJAR EN PANTALLA

Ilustración 61 - Pseudo Código del algoritmo de K-medias.

Ir a:



ROOM FULL OF EMPTYNESS



EL MODELO MATEMÁTICO



VISIÓN COMPUTACIONAL



MÁQUINA AUTOMÁTICA PROGRAMABLE



