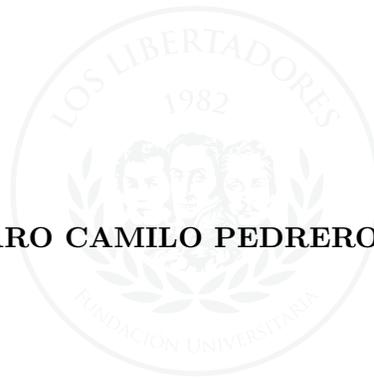


**DISEÑO Y SIMULACIÓN DEL RECONOCIMIENTO DE UN ENTORNO
ROBÓTICO BASADO EN LA DESCOMPOSICIÓN DEL PLANO EN
TESELACIONES USANDO FRENTE DE ONDA CON CUADRADOS**

ALVARO CAMILO PEDREROS BENAVIDES



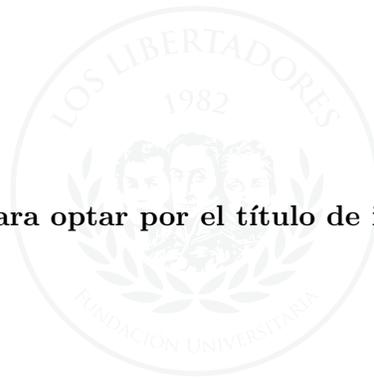
LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
BOGOTÁ DC
2017**

**DISEÑO Y SIMULACIÓN DEL RECONOCIMIENTO DE UN ENTORNO
ROBÓTICO BASADO EN LA DESCOMPOSICIÓN DEL PLANO EN
TESELACIONES USANDO FRENTES DE ONDA CON CUADRADOS**

ALVARO CAMILO PEDREROS BENAVIDES

Trabajo para optar por el título de ingeniero electrónico



LOS LIBERTADORES

FUNDACIÓN UNIVERSITARIA

**Director
Ivan Dario Ladino Vega**

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES

**FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA**

BOGOTÁ DC

2017

Nota de aceptación



Firma
Nombre:
Presidente del jurado

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Firma
Nombre:
Jurado

Firma
Nombre:
Jurado

Bogotá DC, Mayo 22 de 2017

AGRADECIMIENTOS

Por su acompañamiento y guía Ing. Ivan Dario Ladino Vega, Ing. Brayan Jair Saenz Cabezas, Ing. Nelson Eduardo Lozano Espinosa, a mis padres por su paciencia, apoyo y constancia a mi lado, a mis compañeros por su colaboración en la elaboración de esta tesis Diego Fernando Pinzón Patiño, Oscar Fernando Penagos Espinel a Diego Penagos Espinel por su colaboración en la elaboración de las imágenes de este documento, a la fundación universitaria los libertadores por ayudarme a conseguir uno de mis sueños, ser Ing electronico.



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

CONTENIDO

	Pág.
INTRODUCCIÓN	9
OBJETIVOS	11
1. marco teórico	12
1.1. Espacios Topológicos	12
1.2. Topología euclidiana	13
1.3. Base de una topología	14
1.4. Puntos Límites	16
1.5. Homeomorfismos	17
1.5.1. Sub espacios	17
1.5.2. Espacios No Homeomorfos	17
1.6. Espacios Métricos	19
1.7. Topología algebraica	20
1.7.1. Topología de conjuntos de puntos	20
1.7.2. Homotopía, deformación retractada y homotopía equivalente	20
1.7.3. Incentivo de la topología algebraica	23
1.7.4. Operador de frontera	23
1.7.5. Distribución de un operador de frontera y orientación	24

1.7.6.	Construcción de grupos	26
1.7.7.	Descripción de la homología	26
1.7.8.	Estructura de grupo de $H_n(X)$	28
1.7.9.	Interpretación de los grupos homológicos	29
1.7.10.	Geometría Elemental de Riemann	29
1.7.11.	Variedad	30
1.8.	Algoritmo de búsqueda gráfica	31
1.8.1.	Algoritmo de Dijkstra	31
1.8.2.	Algoritmo A^*	33
2.	Implementación	35
2.1.	Creación del entorno	35
2.2.	Discretización del entorno	37
2.3.	Expansión y limitación de las teselaciones	38
2.4.	ubicación de las teselaciones	45
2.5.	Comprobación de ubicación de las teselaciones	48
2.6.	Calculo de centros de masa	51
2.7.	Resultados	54
3.	Conclusiones	57
4.	Anexos	58
	BIBLIOGRAFÍA	59

LISTA DE FIGURAS

		Pág.
1	Figura 1.7.1 : Una deformación de incorporación de X a $A \subseteq X$, para cada t , el área verde es $F(X, t)$	21
2	Figura 1.7.2 : Un cilindro y un toro solido son homotópicos equivalentes. Cada uno de ellos es un homotópico equivalente a un círculo.	22
3	Figura 1.7.3 : Operador limite actuando sobre una 2-cadenas originando una cadena vacía.	23
4	Figura 1.7.4: Distributividad de un operador límite.	24
5	Figura 1.7.5 : Un conjunto de 2-cadenas y su su frontera, con coeficientes representados por colores.	25
6	Figura 1.7.6 : Ciclos en diferente e igual clase homológica.	28
7	Figura 1.7.7: El grado de homología del grupo dá el número de Betti. La clase homológica de un ciclo arbitrario puede expresarse como una combinación lineal de los generadores de los grupos homológicos. $[Z_p]$ y $[Z_q]$ son generadores de $H_1(X)$	30
8	Figura 1.8.1: Representación en diagramas de flujo del algoritmo de Dijkstra.	32
9	Figura 1.8.2: Ilustración del proceso del algoritmo de Dijkstra. Nodo de inicio en rojo.	32
10	Figura 1.8.4: Ilustración del proceso del algoritmo A^* . El conjunto abierto esta marcado por los círculos azules vacíos.	33
11	Figura 1.8.3: Representación en diagramas de flujo del algoritmo de A^*	34
12	Figura 2.1: Entorno aleatorio creado por el algoritmo 1.	36
13	Figura 2.2: Diagrama de flujo de Entorno aleatorio.	36

14	Figura 2.3: Diagrama de Teselaciones a partir de Cuadrados iter = 0.13.	39
15	Figura 2.4: Diagrama de Teselaciones a partir de Cuadrados iter = 0.33.	39
16	Figura 2.5: Diagrama de Teselaciones a partir de Cuadrados iter = 0.66.	40
17	Figura 2.6: Diagrama de Teselaciones a partir de Cuadrados iter = 1.	40
18	Figura 2.7: Diagrama de Teselaciones a partir de Círculos iter = 0.13.	41
19	Figura 2.8: Diagrama de Teselaciones a partir de Círculos iter = 0.33.	41
20	Figura 2.9: Diagrama de Teselaciones a partir de Círculos iter = 0.66.	42
21	Figura 2.10: Diagrama de Teselaciones a partir de Círculos iter = 1.	42
22	Figura 2.11: Diagrama de Teselaciones a partir de rombos iter = 0.13.	43
23	Figura 2.12: Diagrama de Teselaciones a partir de rombos iter = 0.33.	43
24	Figura 2.13: Diagrama de Teselaciones a partir de rombos iter = 0.66.	44
25	Figura 2.14: Diagrama de Teselaciones a partir de rombos iter = 1.	44
26	Figura 2.15: Entorno con teselaciones aleatorias	45
27	Figura 2.16: Entorno con teselaciones ordenadas.	46
28	Figura 2.17: Diagrama de flujo de expansión de teselaciones	47
29	Figura 2.18: Diagrama de área no cubierta.	49
30	Figura 2.19: Entorno con teselaciones ordenadas	50
31	Figura 2.20: Entorno con teselaciones ordenadas con filas desplazadas	50
32	Figura 2.21: centros de masa de teselaciones ordenadas.	51
33	Figura 2.22: Diagrama de flujo de centros de masa de teselaciones	53

INTRODUCCIÓN

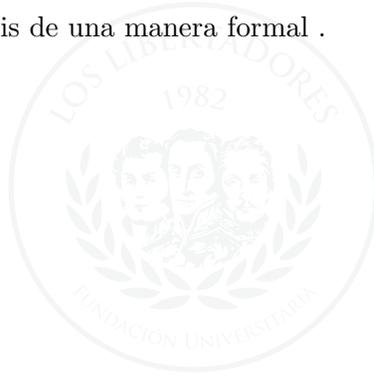
Se han creado maquinas o robots al punto de ser autónomos, pero con un movimiento muy limitado. Durante el siguiente documento se detallará el procedimiento de reconocimiento del entorno por parte de un robot, usando un área de la matemática formal como lo es la topología, reduciendo los recursos necesarios y el tiempo de ejecución frente a los procesos implementados habitualmente, los cuales consisten principalmente en realizar dicha adquisición mediante procesos de ensayo y error.



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

PLANTEAMIENTO DEL PROBLEMA

El hombre ha creado máquinas para hacer diferentes tareas por lo que estas necesitan desplazarse entre diferentes puntos, esto hace necesario que las maquinas tengan una forma para reconocer el entorno que las rodea y desplazarse en este para realizar estas tareas, por lo que se han creado algunos algoritmos para reconocimiento de entornos y desplazamiento en este como lo son: Dijkstra, A*, D*, R* y ARA* junto con otros algoritmos siendo más robustos, teniendo el inconveniente que en algunos casos estos no son eficientes al discretizar el entorno por el cual se van a desplazar, al no lograr que el entorno este bien discretizado las maquinas no logran terminar con sus tareas por lo que pueden tropezar con un obstáculo, no encontrar su objetivo de manera eficiente o precisa y siendo uno de los peores casos no lograr encontrar el objetivo, al no tener discretizado de forma eficiente el entorno por lo que se quiere hacer este análisis de una manera formal .



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

OBJETIVOS

OBJETIVO GENERAL

Discretizar entornos de dos dimensiones que cumplan con la topología euclidiana por medio de teselaciones para la búsqueda de rutas.

OBJETIVOS ESPECÍFICOS

- Utilizar de manera óptima herramientas de topología para la discretización de un entorno.
- Definir el número de teselaciones de acuerdo al tamaño del entorno, la cantidad de obstáculos y el área total que ocupan.
- Determinar la mejor forma de ubicar las teselaciones para cubrir el área total del entorno sin perder información de este.

1. MARCO TEÓRICO

1.1. ESPACIOS TOPOLÓGICOS

Empezando con la definición general, el término topología se utiliza para identificar a un área de la matemática que estudia la continuidad y otros conceptos originados a partir de ella. Se trata de una especialización vinculada a las propiedades y características que poseen los cuerpos geométricos y que se mantienen sin alteraciones gracias a cambios continuos, con independencia de su tamaño o apariencia.

Definición 1.1 Sea X un conjunto no vacío. Una colección \mathcal{T} de subconjuntos de X se dice que es una topología sobre X si:

- X y el conjunto vacío, \emptyset , pertenecen a \mathcal{T} .
- La unión de cualquier número finito o infinito de conjuntos en \mathcal{T} pertenecen a \mathcal{T}
- La intersección de dos conjuntos cualesquiera de \mathcal{T} pertenece a \mathcal{T}

Definición 1.2 Sea X cualquier conjunto no vacío, y \mathcal{T} la colección de todos los subconjuntos de X . Entonces \mathcal{T} es llamada topología discreta sobre el conjunto X . El espacio topológico (X, \mathcal{T}) se llama espacio discreto.

Definición 1.3 Sea X un conjunto no vacío cualquiera y $\mathcal{T} = (X, \emptyset)$. Entonces \mathcal{T} es conocido como topología indiscreta y (X, \mathcal{T}) se dice que es un espacio indiscreto.

Así como los espacios topológicos son diferenciables, se puede definir los miembros de las topologías.

1.2. TOPOLOGÍA EUCLIDIANA

Esta dada por el conjunto de los números reales, denotados mediante \mathbb{R} , en el cual si se toma un subconjunto S de \mathbb{R} que cumpla con el siguiente enunciado, y sea abierto, se consideraría en la topología euclidiana sobre \mathbb{R} .

- Por cada $x \in S$, existen a, b en \mathbb{R} , con $a < b$, tal que $x \in (a, b) \subseteq S$

Siempre que se refiere al espacio topológico \mathbb{R} sin especificar la topología, se asume \mathbb{R} como la *topología euclidiana*.

Sin dejar de lado la sección euclidiana; en particular, se verá que todos los intervalos abiertos son en efecto conjuntos abiertos en esta topología y que todos los intervalos cerrados son conjuntos cerrados.

Se a de tener en cuenta, adicional a lo anterior, que mientras cada intervalo abierto es un conjunto abierto en \mathbb{R} , lo contrario no siempre lo es. No todos los conjuntos abiertos en \mathbb{R} son intervalos.

Un ejemplo del caso mencionado anteriormente sería: $(1, 3) \cup (5, 6)$ en un conjunto abierto en \mathbb{R} , pero no es un intervalo abierto.

Definición 1.2.1 Sea $(X, \mathcal{T}$ un espacio topológico. Una colección β de subconjuntos abiertos de X se llama **Base** de una topología \mathcal{T} si cada conjunto abierto es una unión de miembros de β .

Es decir, que si β es una base de una topología \mathcal{T} sobre un conjunto X entonces un subconjunto U de X está en \mathcal{T} si, y solo si, es una unión de miembros de β . Por lo que β "genera" la topología \mathcal{T} en el sentido siguiente: Si se informa cuales conjuntos son miembros de β , entonces se podrá determinar los miembros de \mathcal{T} , ellos son todos los conjuntos que son uniones de miembros de β .

Si X es un conjunto no vacío y β es una colección de subconjuntos de X . Entonces β es una base de una topología sobre X si, y solo si, β tiene ciertas propiedades:

- $X = \bigcup_{\beta \in \beta} \beta$.

- Para cualesquiera $\beta_1, \beta_2 \in \beta$, el conjunto $\beta_1 \cap \beta_2$ es una unión de miembros de β .

Se ha definido un espacio topológico muy importante \mathbb{R} , el conjunto de todos los números reales con la topología euclidiana; en esta topología, los intervalos abiertos son ciertamente conjuntos abiertos (y los intervalos cerrados son conjuntos cerrados). Sin embargo, no todos los conjuntos abiertos son intervalos abiertos. No obstante, todo conjunto abierto en \mathbb{R} es una unión de intervalos abiertos. Ésto llevó a introducir la noción de "base de una topología" y establecer que la colección de todos los intervalos abiertos es una base de la topología Euclidiana sobre \mathbb{R} .

"La topología euclidiana " \mathcal{T} es una topología: Para que esto se cumpla \mathcal{T} tiene que cumplir con las definiciones de la topología básica

- Sean $r, s \in \mathbb{R}$ como $r < s$ En la topología euclidiana \mathcal{T} sobre Topología \mathbb{R} el intervalo abierto (r, s) ciertamente pertenecen a \mathcal{T} , siendo por lo tanto un conjunto abierto.
- Los intervalos abiertos (r, ∞) y $(-\infty, r)$ son conjuntos abiertos en \mathbb{R} , para cada número real r .
- Es importante notar que mientras cada intervalo abierto es un conjunto abierto en \mathbb{R} lo contrario es falso. No todos los conjuntos abiertos en \mathbb{R} son intervalos. Por ejemplo el conjunto $(1, 3) \cup (5, 6)$ es un conjunto abierto en \mathbb{R} , pero no es un intervalo abierto. Incluso el conjunto $\bigcup_{n=1}^{\infty} (2n, 2n + 1)$ es un conjunto abierto en \mathbb{R}
- Para cada c y d en \mathbb{R} con $c < d$, el intervalo cerrado $[c, d]$ no es un conjunto abierto en \mathbb{R}
- Por cada a y b en \mathbb{R} con $a < b$, el intervalo cerrado $[a, b]$ es un conjunto cerrado en la topología euclidiana sobre \mathbb{R} .
- Cada conjunto unitario $\{a\}$ es cerrado en \mathbb{R}
- El conjunto \mathbb{Q} de todos los números racionales no es un subconjunto cerrado de \mathbb{R} ni es un subconjunto de \mathbb{R}

1.3. BASE DE UNA TOPOLOGÍA

Las observaciones anteriores permiten describir la topología euclidiana sobre \mathbb{R} de una manera más conveniente para esto se ha de tener clara la noción base de una topología .

Proposición 1.3.1 Un subconjunto S de \mathbb{R} es abierto si y solo si es la unión de intervalos abiertos.

Tenemos que demostrar que S es abierto si y sólo si, es la unión de intervalos abiertos; es decir

- si S es una unión de intervalos abiertos, entonces S es un conjunto abierto, y
- si S es un conjunto abierto, entonces S es la unión de intervalos abiertos.

La proposición anterior se dice que para describir la topología de \mathbb{R} basta decir que todos los intervalos (a, b) son conjuntos abiertos, y que todos los otros conjuntos abiertos son una unión de estos conjuntos abiertos. Ésto lleva a la definición siguiente.

Definición 1.3.1 Sea (X, \mathcal{T}) un espacio topológico. Una colección β de subconjuntos abiertos de X se llama base de una Topología \mathcal{T} si cada conjunto abierto es una unión de miembros de β . Si β es una base de una topología \mathcal{T} sobre un conjunto X entonces un subconjunto \cup de X está en \mathcal{T} si y sólo si, es una unión de miembros de β . Por lo que β "genera" la topología \mathcal{T} en el sentido siguiente: si se informa cuáles conjuntos son miembros de \mathcal{T} entonces poder determinar los miembros de \mathcal{T} ellos son todos los conjuntos que son uniones de miembros de \mathcal{T} .

Observación. si (X, \mathcal{T}) es un espacio topológico entonces $\beta = \mathcal{T}$. es una base de la topología \mathcal{T} . Por lo que, por ejemplo, el conjunto de todos los subconjuntos de X es una base de la topología discreta sobre X .

Pueden existir muchas bases diferentes para la misma topología. En efecto, si β es una base de una topología \mathcal{T} . Sobre un conjunto X y β_1 es una colección de subconjuntos de X tal que $\beta \subseteq \beta_1 \subseteq \mathcal{T}$, entonces β_1 es también una base de \mathcal{T} . La noción de "base de una topología" permite definir topologías. Sin embargo, el ejemplo siguiente muestra que se tendrá cuidado.

Proposición 1.3.2 Sea X un conjunto no vacío y sea β una colección de subconjuntos de X . entonces β es una base de una Topología sobre X si y solo si, β tiene las propiedades siguientes:

- $X = \cup_{B \in \beta} B$
- para cualesquier $B_1, B_2 \in \beta$, el conjunto $B_1 \cap B_2$ es una unión de miembros de β

1.4. PUNTOS LÍMITES

Para esta sección, se tratarán los temas de proximidad, de forma específica, defini la función de punto límite sin llegar a recurrir a distancias; la cual ayudará a comprender mejor las nociones de conjunto cerrado. Otro tema tratado es el de "conexidad" (Intuitivamente, un conjunto conexo es aquel formado por una sola "pieza", que no se puede "dividir").

Definición 1.4.1 Sea A un subconjunto de un espacio topológico (X, \mathcal{T}) . Un punto $x \in X$ se llama punto límite (o punto de acumulación o punto de contacto) de A si todo conjunto abierto U que contiene x contiene un punto A diferente de x .

En donde si (X, \mathcal{T}) es un espacio topológico, es usual referirse a los elementos del conjunto X como **puntos**. Para entender y aterrizar este concepto, se tratará con un ejemplo.

Proposición 1.4.1 Sea A es un subconjunto de un espacio topológico (X, \mathcal{T}) . Entonces A es cerrado en (X, \mathcal{T}) si, y solo si, A contiene todos sus puntos y límites.

Proposición 1.4.2 Sea A es un subconjunto de un espacio topológico (X, \mathcal{T}) y A' el conjunto de todos los puntos límites de A . Entonces $A \cup A'$ es un conjunto cerrado.

Definición 1.4.2 Sea A es un subconjunto de un espacio topológico (X, \mathcal{T}) , entonces $A \cup A'$ que consiste en A y todos sus puntos límites, se llama clausura de A y se denota por \bar{A} .

1.5. HOMEOMORFISMOS

En cada rama de las matemáticas resulta esencial reconocer cuando dos estructuras son equivalentes. Por ejemplo, dos conjuntos son equivalentes, en teoría de conjuntos, si existe una función biyectiva que transforma un conjunto en el otro. Dos grupos son equivalentes (o isomorfos) si existe un homomorfismo de uno al otro que es inyectivo y sobreyectivo. Dos espacios topológicos son equivalentes (u homeomorfos), si existe un homeomorfismo de uno sobre el otro.

1.5.1. Sub espacios

Definición 1.5.1.1 Sea Y un sub conjunto no vacío de un espacio topológico (X, \mathcal{T}) La colección $\mathcal{T}_Y = \{O \cup Y : O \in \mathcal{T}\}$ de subconjuntos de Y es una topología sobre Y llamada la topología del subespacio (o la topología relativa o la topología inducida o la topología inducida sobre Y por \mathcal{T}). El espacio topológico (Y, \mathcal{T}_Y) se llama un subespacio de (X, \mathcal{T}) .

1.5.2. Espacios No Homeomorfos

Para demostrar que dos espacios topológicos son homeomorfos, se ha de encontrar un homeomorfismo entre ellos. Dos espacios topológicos no son homeomorfos es con frecuencia mucho más difícil, tener que mostrar que no existe ningún homeomorfismo. El ejemplo siguiente da una pista de cómo se puede proceder en este caso.

Proposición 1.5.2.1 cualquier espacio topológico homeomorfo a un espacio conexo es conexo. La Proposición brinda una vía para tratar de probar que dos espacios topológicos no son homeomorfos encontrando una propiedad "preservada por homeomorfismos" que un espacio tiene y el otro no.

Con lo visto muchas propiedades son "preservadas por homeomorfismos" como son :

- 1 T_0 -espacio
- 2 T_1 -espacio
- 3 T_2 -espacio o Hausdorff espacio
- 4 espacio regular
- 5 T_3 -espacio

- 6 satisfacer el segundo axioma de numerabilidad
- 7 espacio separable.

Y también otras :

- 1 espacios discreto
- 2 espacio indiscreto
- 3 topología cofinita
- 4 topología numerable cerrada

Estas con propiedades preservadas por homeomorfismos. Además de esto, dos espacios (X, \mathcal{T}) y (Y, \mathcal{T}_1) no pueden ser homeomorfos si X y Y tienen cardinalidades diferentes. No obstante, cuando se enfrenta a un problema específico, pudiera no tener la propiedad que se necesita.

Definición 1.5.2.1

Un subconjunto S de \mathbb{R} se dice que es un intervalo si este tiene la propiedad siguiente:

si $x \in S, z \in S,$ y $y \in \mathbb{R}$ son tales que $x < y < z$, entonces $y \in S$

Observaciones Note que cada conjunto unitario $\{x\}$ es un intervalo.

- 1 Note que cada conjunto unitario $\{x\}$ es un intervalo.
- 2 cada intervalo tiene una de las formas siguientes:
 $\{a\}, [a, b], (a, b), [a, b), (a, b], (-\infty, a), (-\infty, a], (a, \infty), [a, \infty), (-, \infty), \infty)$
- 3 Cada intervalo es homeomorfo a $(0, 1), [0, 1], [0, 1)$ o $\{0\}$

1.6. ESPACIOS MÉTRICOS

La clase más importante de espacios topológicos es la clase de espacio métrico. Los espacios métricos constituyen una fuente rica de ejemplos en topología. Pero más que esto, la mayoría de las aplicaciones de la topología al análisis son a través del espacio métrico.

La noción de espacio métrico fue introducida en 1906 por Maurice Fréchet, y fue desarrollada y acuñada por Félix Hausdorff en 1914.

Definición 1.6.1 (Espacios métricos). Sea X un conjunto no vacío y d una función de valor real definida sobre $X \times X$ tal que para $a, b \in X$:

- $d(a, b) \geq 0$, y $d(a, b) = 0$ si, y sólo si, $a = b$;
- $d(a, b) = d(b, a)$
- $d(a, c) \leq d(a, b) + d(b, c)$, para toda a, b y c en X [la desigualdad triangular].

Entonces d es llamada métrica sobre X . (X, d) es llamado espacio métrico y $d(a, b)$ se conoce como la distancia entre a y b .

Definición 1.6.2. Sean (X, d) un espacio métrico y r un número real positivo cualquiera. Entonces la bola abierta centrada en $a \in X$ y de radio r es el conjunto $B_r(a) = \{x : x \in X \text{ y } d(a, x) < r\}$.

Proposición 1.6.2 Sea (X, d) un espacio métrico. Entonces la colección de bolas abiertas en (X, d) es una base de una topología \mathcal{T} sobre X .

[la topología \mathcal{T} se conoce como la topología inducida por la métrica d , y (X, \mathcal{T}) se conoce como el espacio topológico inducido o el espacio topológico correspondiente o el espacio topológico asociado.

Definición 1.6.3 Métricas sobre un conjunto X se llaman equivalentes si ellas inducen la misma topología sobre X .

1.7. TOPOLOGÍA ALGEBRAICA

1.7.1. Topología de conjuntos de puntos

La teoría de conjuntos es el estudio de colecciones para algunos objetos. En la mayoría de los casos la colección puede ser infinita o no contable. Por ejemplo, uno puede hablar acerca del conjunto de todos los puntos de la superficie de una esfera. Además, la teoría de conjuntos hace una pequeña relación entre objetos de un conjunto. Por ejemplo, si considera el conjunto que consiste en los puntos de una esfera, esto no es una colección de puntos, cada uno es distinto y no hay forma de afirmar que un punto está conectado con algún otro. Ahí es donde entra la topología. La topología consiste en un conjunto, junto con información adicional sobre “agrupación”/”recolección” de los objetos dentro del conjunto. Las agrupaciones son llamadas subconjuntos abiertos del conjunto.

1.7.2. Homotopía, deformación retractada y homotopía equivalente

La idea de deformación retractada es tener un espacio topológico X y un sub-espacio A (Un subconjunto con sub-espacio topológico), y preguntar si el espacio X puede ser continuamente ‘encogido’ o ‘deformado’ en A sin causar alguna ‘cortadura’ o ‘lagrima’. Si esto se puede, llamar A a retraer la deformación de X (Figura 1.7.1). Se consideró el mapa de identidad $\text{id}_X : X \rightarrow X$ (Figura 1.7.1 (a)). Ahora se empezó la ‘contracción’ de X gradualmente a ‘colapsar’ en A . Para cada paso del proceso de contracción se obtuvo una incrustación de X en si mismo, tal que la imagen de la incrustación es la versión ‘encogida’ de X a cada paso (Figura 1.7.1 (b,c)). Eventualmente ‘encoger’ X a A (Figura 1.7.1 (d)).

Definición 1.7.2.1 (Deformación retractada [9], Figura 1.7.1). Un sub-espacio A (con un sub-espacio topológico) es llamado una deformación retractada del espacio topológico X si ahí existe una función continua $F : X \times [0, 1] \rightarrow X$ tal que

- (a) $F(x, 0) = x, \forall x \in X (F(., 0) \equiv \text{id}_X$ es el mapa identidad en X)
- (b) $F(a, t) = a, \forall a \in A, t \in [0, 1]$
- (c) $F(x, 1) \in A, \forall x \in X$

F (la cual puede no ser única) es llamada una deformación de retracción de X a A . Ya que A es un sub-espacio de X , interpretar F como una homotopía entre el mapa identidad

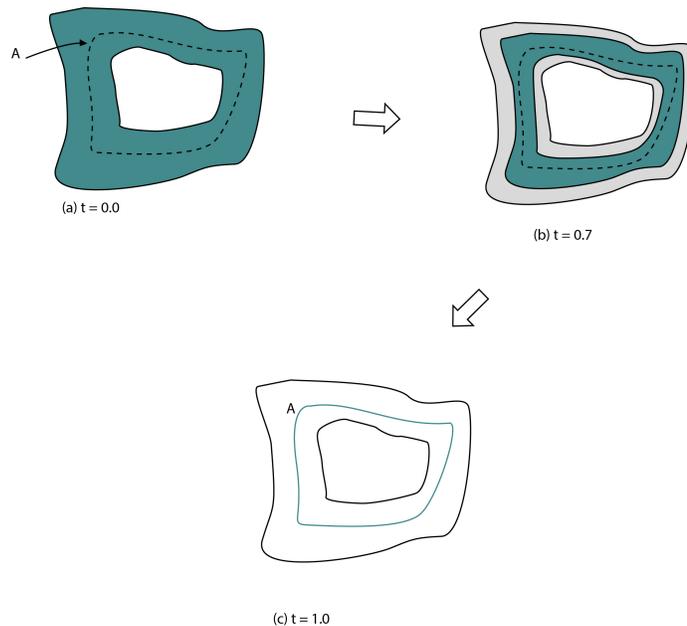


Figura 1.7.1 : Una deformación de incorporación de X a $A \subseteq X$, para cada t , el área verde es $F(X, t)$.

fuelle (Diego Penagos)

- (a) $t = 0.0$.
- (b) $t = 0.4$.
- (c) $t = 1.0$.

id_X y el mapa $f_1 \equiv F(., 1)$ cuya imagen esta en A . Es importante tener en cuenta que $f_1 \equiv F(., 1) : X \rightarrow X$ es homotópico al mapa identidad sobre X . A no ha sido dado de antemano, y en su lugar, una función $f_1 : X \rightarrow X$ la cual es homotópica a id_X , la imagen de f_1 puede ser claramente una deformación retractada de X . El hecho es que A necesita ser un sub-espacio de X en la definición de deformación de retracción, especialmente implica que ahí hay una incrustación $i : A \rightarrow X$, llamada la inclusión. Además A , es un espacio topológico independiente, no debería requerir una incrustación en X para ser descrita (A en la Figura 1.7.1 es topológicamente un círculo \mathbb{S}^1). Es por eso que generaliza una deformación de retracción, llamada homotopía equivalente.

El objetivo de la homotopía equivalente es en lugar de mencionar explícitamente un sub-espacio A de X , mira las funciones continuas desde X hacia ella misma a través de un

segundo espacio Y . (la imagen final es por supuesto, un sub-espacio de X) Luego se pregunta si esa combinación de mapas es homotópico al mapa identidad de X . Con el rol de X y Y invertidos. Si la respuesta es ‘Si’ en los dos casos, los espacios X y Y son conocidos como homotópicos equivalentes.

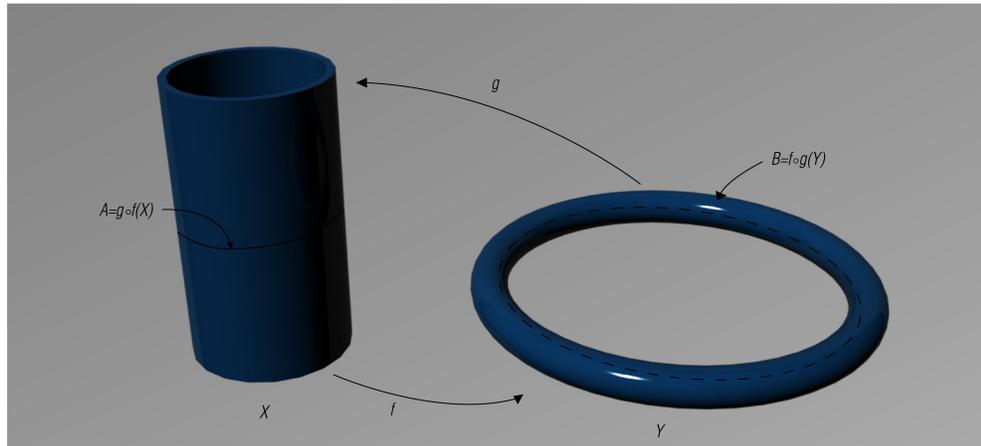


Figura 1.7.2 : Un cilindro y un toro sólido son homotópicos equivalentes. Cada uno de ellos es un homotópicamente equivalente a un círculo.

fuentes (Diego Penagos)

Definición 1.7.2.2 (Equivalencia homotópica [9], Figura 1.7.2). Dos espacios topológicos X y Y son llamados *equivalentes homotópicos* si ahí existe funciones continuas $f : X \rightarrow Y$ y $g : Y \rightarrow X$ tal que $g \circ f$ sea homotópico al mapa identidad id_X , y $f \circ g$ sea homotópico al mapa identidad id_Y . La función f (e igualmente g) es llamada una *homotopía equivalente*. X y Y tienen el mismo tipo homotópico, e informalmente decir que uno puede ser homotópico al otro.

Si A es una deformación retractada de X , entonces ellos son homotópicos equivalentes. Aunque no todas las veces es verdadero. Fuera de muchos caminos de determinación si dos espacios X y Y son homotópicos equivalentes, uno se aproxima a comprobar si cada uno de ellos está retraído a la deformación de un sub-espacio que es topológicamente equivalente (homeomórficos). Entonces ellos son homotópicamente equivalentes (Figura 1.7.2). La otra aproximación más formal es comprobar si existe un espacio más grande con incrustaciones de

X y Y en él, así el espacio más grande será una deformación de retracción de los dos (XyY).

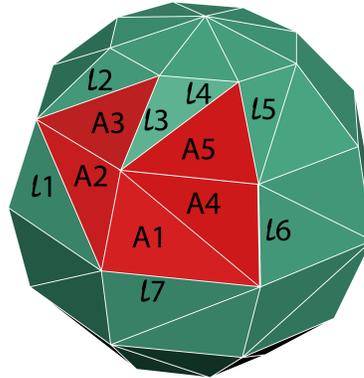


Figura 1.7.3 : Operador limite actuando sobre una 2-cadenas originando una cadena vacía.
fuente (Diego Penagos)

1.7.3. Incentivo de la topología algebraica

El primer paso es impartir una estructura algebraica a un espacio topológico, esto describe el espacio en términos de una secuencia de grupos (en casos sencillos, espacios vectoriales, que son en sí mismos grupos con estructuras adicionales), y mapas entre ellos. Este objeto algebraico se llamara una cadena compleja. Mientras no sea necesario discretizar un espacio topológico para describir una cadena compleja en él, Cada elemento discreto en la discretización es llamado un simplicial (Figura 1.7.3) – los vértices serán llamados 0-simpliciales, los bordes 1-simpliciales, los triángulos 2-simpliciales, los tetraedros 3-simpliciales, etc. Formalmente, un n -simplicial en un espacio topológico, X , es un mapa desde un n -simplicial estándar a X . Además, se referencia en una imagen de un n -simplicial como el n -simplicial de sí mismo.

1.7.4. Operador de frontera

Considere una trayectoria de área sobre un plano que es discretizado en simpliciales como en la Figura 1.7.3. Fuera de todos los 2-simpliciales (triángulos), al coger unos pocos – unos marcados por color rojo como en la Figura 1.7.3. simplemente llamar a estos triángulos A_1, A_2, \dots, A_5 (cabe resaltar que por A_i no se refiere al ‘área’, pero todo el triangulo es un conjunto abstracto). Así la región que cubren es denotada por la frontera e igualmente

representada por $\sum_{i=1}^7 l_i$ (cada 1-simplicial o borde es etiquetado arbitrariamente l_i). Además, si ahora se mira la frontera de $\sum_{i=1}^7 l_i$, esta es claramente vacía (en general, así la frontera puede haberse formado por 0-simpliciales o vértices). Esta última observación es una clave tras la construcción de una cadena compleja. Esta observación se extiende a dimensiones más altas o también a un espacio topológico. Por ejemplo, en un espacio discretizado de tres dimensiones dentro de un tetraedro, si se tomo pocos de esos tetraedros (3-simpliciales) para definir un volumen, y tomar el límite de ese volumen (que será una superficie cerrada), ese límite tendrá de por sí una frontera vacía. Así, el límite de una frontera esta siempre vacío. En una notación ingenua, si $\delta_2(A)$ representa la frontera del área A , y $\delta_1(l)$ representa la frontera de la curva L , se puede resumirlo como $\delta_2(\sum_{j=1}^5 A_j) = \sum_{i=1}^7 l_i$, y $\delta_1(\sum_{i=1}^7 l_i) = 0 = \delta_1 \circ \delta_2(\sum_{j=1}^5 A_j)$. En general, $\delta_n \circ \delta_{n+1} = 0$.

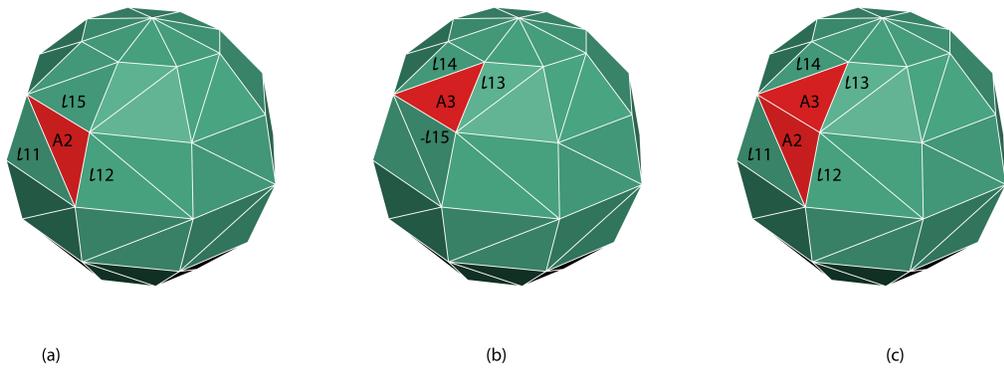


Figura 1.7.4: Distributividad de un operador límite.
fuente (Diego Penagos)

- (a) En la frontera de A_2 se tiene un borde llamado l_{12} .
- (b) El límite de A_3 necesita tener $-l_{12}$ para ser un borde. se puede llamarlo l_{21} , pero esto necesitará ser igual a $-l_{12}$ para asegurar la distributividad de δ_2 .
- (c) Una vez sumado los límites de A_2 y A_3 , los bordes l_{12} y $-l_{12}$ se cancelan, y se obtiene la frontera de $A_2 + A_3$.

1.7.5. Distribución de un operador de frontera y orientación

Una de las propiedades que se aplica para el operador de frontera, δ_n , es tener la propiedad distributiva. Esto se debe a que, por ejemplo, al ser capaces de escribir $\delta_2(A_i + A_j) = \delta_2(A_i) + \delta_2(A_j)$. Esto permitirá afirmar que el límite de la frontera es vacío, puramente por

las condiciones algebraicas, sin mirar la imagen: $\delta_1 \circ \delta_2 \sum_j A_i = \sum_j \delta_1 \circ \delta_2(A_i) = \sum_j 0 = 0$ (hasta el límite de la frontera de un triángulo individual es siempre vacío). Esto habilitaría el desarrollo en álgebra lineal. Esto requiere asignar una dirección a cada l_i . Se considera un 2-simplicial, A_2 , mostrado en la Figura 1.7.4 (a), A_2 y A_3 comparten un mismo borde, l_{12} , que se encuentra dentro de $A_2 + A_3$, se necesita asegurar de algún modo que ese borde sea anulado cuando se suma $\delta_2(A_2)$ a $\delta_2(A_3)$ para obtener $\delta_2(A_2 + A_3)$ (Figura 1.7.4 (c)). Esto se consigue gracias a darle una dirección a cada segmento l_i , representado como $\pm l_i$, y señalando que $l_i + (-l_i) = 0$. Acá no hay nada especial acerca de una dimensión de los 1-simpliciales, y se puede asignar direcciones a los simpliciales de cada dimensión (vértices, bordes, triángulos, tetraedros, etc.). La definición de dirección tiene que ser tal que admita operaciones distributivas de los operadores de frontera δ_n .

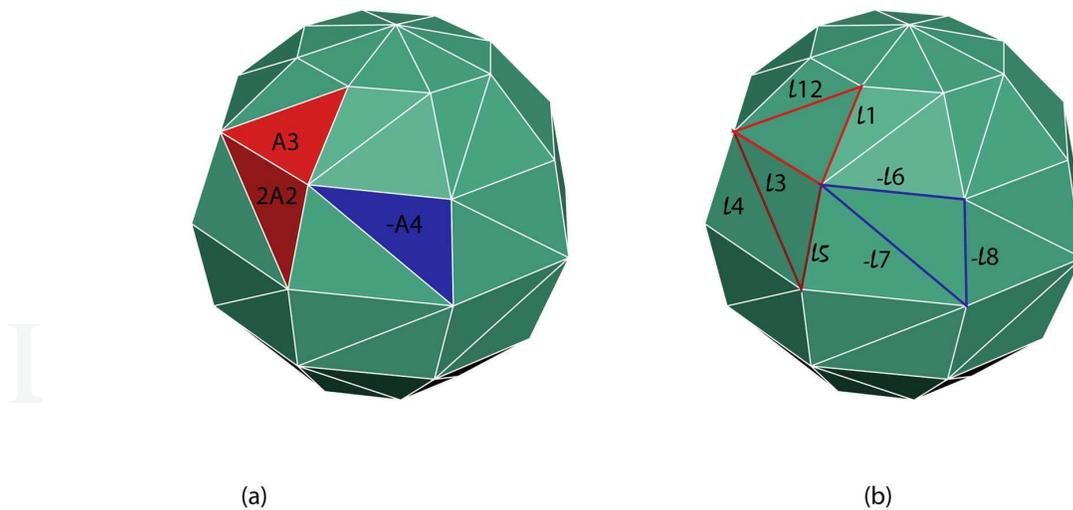


Figura 1.7.5 : Un conjunto de 2-cadenas y su su frontera, con coeficientes representados por colores.

fuelle (Diego Penagos)

- (a) Una representación gráfica de $A_3 + 2A_2 + (-A_4) \in C_2(X)$. Los colores representan los coeficientes enteros - rojo : positivo, azul : negativo. Este tipo de combinación arbitraria se llama una cadena (2-cadenas para este caso). Así, se puede representar como un conjunto de coeficientes ordenados $[0, 2, 1, -1, 0, \dots]$ para cada 2-simpliciales correspondiente.
- (b) El límite de 2-cadenas está mostrado en (a), el límite es $l_1 + l_2 + l_3 + 2l_4 + 2l_5 - l_6 - l_7 - l_8$.

Puesto que ahora se puede obtener coeficientes arbitrarios enteros, la representación por colores es el camino preferido para su visualización.

1.7.6. Construcción de grupos

Por ahora es fácil ver surgir estructura de grupos. Por ejemplo, por cada segmento de línea l_i , se debe definir un elemento inverso, $-l_i$, que al sumarlos, el resultado sea cero, el elemento de identidad. También, se debe desarrollar la intuición de cómo el operador binario '+' trabaja entre l_i y l_j para $i \neq j$ o $l_j = -l_i$. Todo lo que ahora se necesita hacer es el conjunto de las posibles combinaciones de 1-simpliciales ($l_{i1} + l_{i2} + \dots$ en una combinación arbitraria llamada una 1-cadena) un grupo algebraico es *cerrado* bajo el elemento de adición. Fácilmente se relaciona $l_i + l_j$ tomando la uniones de los segmentos de línea l_i y l_j . Además, si se escribe $l_i + l_j$, una interpretación en términos de unión, significara únicamente l_i . Esto no será consistente con nuestro intento de definir un grupo. Así, se define un nuevo elemento $2l_i := l_i + l_i$. Esto puede ser interpretado tomando un segmento de línea dos veces arriba del (similar a la división de la unión). Además, esto puede ser tenido en cuenta que es una construcción absolutamente algebraica. Siguiendo a lo largo de líneas similares, se puede definir $2l_i, 3l_i, 4l_i, \dots$ y sus correspondientes inversas $-2l_i, -3l_i, -4l_i, \dots$. En general $nl_i = l_i + l_i + \dots (n \text{ veces})$, y $-ml_i = -l_i - l_i + \dots (m \text{ veces})$. Así se debe construir un grupo abeliano que es generado libremente por l_1, l_2, l_3, \dots representar este grupo por $C_1(X)$ (Donde, X es un espacio topológico que se discretiza para la creación de los simpliciales), donde el sub-índice 1 se refiere a la dimensión de los simpliciales. Desde luego se puede hacer un procedimiento similar para los simpliciales de todas las dimensiones (Figura 1.7.5). El grupo para los simpliciales de n-dimensiones son escritos como $C_n(X)$.

1.7.7. Descripción de la homología

Una vez al tener establecido el objetivo de definir cadenas complejas, se podrá formalmente definir la homología como lo siguiente.

Definición 1.7.7.1. Una cadena compleja es una secuencia de grupos abelianos, $\dots, C_3, C_2, C_1, C_0, C_{-1}, \dots$, a lo largo con homomorfismos $\delta_n : C_n \rightarrow C_{n-1}$, tal que $\delta_{n-1} \circ \delta_n = 0$ para todo $n = \dots, 3, 2, 1, 0, -1, \dots$. Esto es comúnmente representado usando el siguiente diagrama:

$$\dots \longrightarrow C_{n+3} \xrightarrow{\delta_{n+3}} C_{n+2} \xrightarrow{\delta_{n+2}} C_{n+1} \xrightarrow{\delta_{n+1}} C_n \xrightarrow{\delta_n} C_{n-1} \xrightarrow{\delta_{n-1}} \dots$$

$$\text{con, } \delta_{n-1} \circ \delta_n = 0, \forall n.$$

Tenga en cuenta que en general, las cadenas complejas no necesitan ser relacionados a algún espacio topológico, como en la definición independiente indicado anteriormente. Esto es simplemente una secuencia de grupos abelianos C_\bullet con los operadores δ_\bullet (por sub-índice ' \bullet ' informalmente se refiere a la colección para todos n). Tal estudio independiente de cadenas complejas sin ninguna referencia a topología es conocida como álgebra homológica y es un campo de estudio por derecho propio. La topología algebraica toma prestada una cantidad significativa de herramientas de este campo.

En topología algebraica, ese grupos C_n corresponden obviamente a los grupos libremente generados por el n -simpliciales en el espacio topológico X , y son escritos como $C_n(X)$. Las cadenas complejas generadas por un número finito de simpliciales como se describe anteriormente son conocidas como *complejos simpliciales*. Sin embargo existen otras, y formas más generales de cadenas complejas que uno puede definir en un espacio topológico, los complejos Δ , complejos singulares, complejos celulares, complejos cúbicos, etc.

Ya que $\delta_n \circ \delta_{n+1} = 0$

$$\text{Img}(\delta_{n+1}) \subseteq \text{Ker}(\delta_n) \subseteq C_n(X)$$

Definición 1.7.7.2. (Sub-grupo de límites). $B_n(X) := \text{Img}(\delta_{n+1}) \subseteq C_n(X)$ es llamado el grupo de n -límites (un sub-grupo de C_n), y es la imagen del todo de $C_{n+1}(X)$ bajo la acción de δ_{n+1} . Elementos en $B_n(X)$ (llamado n -límites) son n -cadenas, cada una de las cuales son límites de las $(n - 1)$ -cadenas en X (Figura 1.7.6(b)).

Definición 1.7.7.3. (Sub-grupo de ciclos). $Z_n(X) := \text{Ker}(\delta_n) \subseteq C_n(X)$ es llamado el grupo de n -ciclos (un sub-grupo de C_n), y es el *núcleo* de δ_n (todos los elementos en C_n que se asignan a la identidad elemental en $C_{n-1}(X)$ bajo la acción de δ_n , Figura 1.7.6(a)). Claro que todos los n -límites son n -ciclos también, pero la conversión no es verdadera.

1.7.8. Estructura de grupo de $H_n(X)$

Para un determinado $z \in Z_n(X)$, escribir $[z] \in H_n(X)$ para la clase homológica de Z . El operador de adición de $H_n(X)$ es heredado de $Z_n(X)$ de una forma bastante natural - definir:

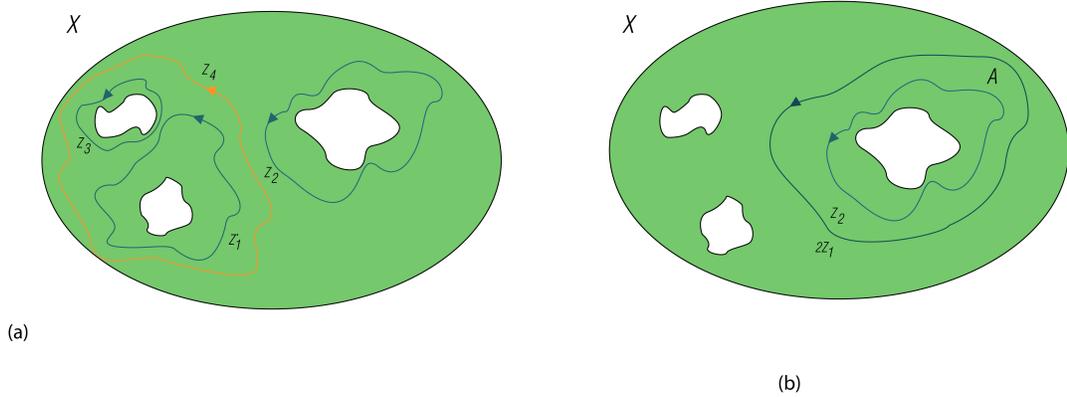


Figura 1.7.6 : Ciclos en diferente e igual clase homológica.

fuente (Diego Penagos)

- (a) z_1' y z_2 son 1-ciclos que están en **diferente** clase homológica. Esto es porque uno no puede encontrar un conjunto 2-ciclo A tal que $z_1 - z_2 = \delta_2(A) \in B_n(X)$.
- (b) $2z_1$ (donde z_1 es el mismo utilizado en (a)) y z_2 son 1-ciclos los cuales están en **diferentes** clases homológicas. El coeficiente 2 para z_1 (el cual puede ser pensado como dos copias de z_1 se colocaron una encima de otra) está indicado por el color oscuro (referido en la Figura 1.7.5(b)).

$[z_1] + [z_2] = [z_1 + z_2]$, donde la primera adición es la uno en $H_n(X)$, mientras la adición en la derecha es conocida por $Z_n(X)$. También, la identidad elemental (o 'cero') de $H_n(X)$ es la clase homológica de los límites (elementos de $B_n(X)$). Esto puede ser visto de la siguiente forma: Si z_1 y \bar{z}_1 pertenecen a la misma clase homológica ($\bar{z}_1 = z_1 + b$ para algún caso $b \in B_n(X)$), entonces por definición, $[z_1] = [\bar{z}_1] \Rightarrow [z_1 + b] = [z_1] \Rightarrow [z_1] + [b] = [z_1] \Rightarrow [b] = 0$.

Recurriendo brevemente a nuestra comparación anterior de $C_n(X)$ con un espacio vectorial, poder ver que $Z_n(X)$ es como un sub-espacio vectorial (un espacio vectorial por derecho propio).

Así, $B_n(X)$ es un sub-espacio de $Z_n(X)$.

Entonces $H_n(X)$ podría ser visto como un sub-espacio vectorial de $Z_n(X)$ el cual es ortogonal

para $B_n(X)$ de tal manera que $H_n(X)$ y $B_n(X)$ se extiende por la totalidad de $Z_n(X)$.

1.7.9. Interpretación de los grupos homológicos

La interpretación del grupo homológico n -ésimo, $H_n(X)$, como se acaba de decir, representa una clase de n -ciclos que difieren en n -fronteras. Esto es además, una interpretación mas clara y intuitiva de los grupos homológicos - el rango (o informalmente, su dimensión) de los grupos se dice acerca del numero Betti n -ésimo (informalmente, el numero de $(n + 1)$ - 'agujeros' dimensionales cuando $n > 0$, y el numero de componentes conectados cuando $n = 0$) de un espacio topológico X . Esto no es difícil de ver del ejemplo en la Figura 1.7.6. Uno puede ver que el de los dos agujeros en el espacio X , acá hay dos tipos distintos de ciclos que no son fronteras (llamados ciclos no triviales) - uno puede ir alrededor del agujero derecho y otro que va alrededor del agujero izquierdo (Figura 1.7.7). Acá hay generadores de $H_1(X)$. De hecho una cálculo directo de $H_1(X)$ revela que este es isomórfico al grupo de \mathbb{Z}^2 (suma directa de dos copias de grupos enteros) - así, el primero número Betti del espacio B1, es 2.

1.7.10. Geometría Elemental de Riemann

Topología, debido a esta invariante a homeomorfismo, no dice acerca de la distancia entre puntos sobre un espacio topológico. Uno necesita definir una función de medición de distancia sobre un espacio topológico para ello.

Definición 1.7.10.1 (Función de distancia [11]). Una función de distancia sobre un espacio topológico X es una función $d : X \times X \rightarrow \mathbb{R}$ tal que, para algún $p, q \in X$, las siguientes condiciones son satisfechas:

- (a) $d(p, q) \geq 0$.
- (b) $d(p, q) = 0$ si y solo si $p = q$.
- (c) $d(p, q) = d(q, p)$.
- (d) $d(p, q) \leq d(p, r) + d(r, q)$ para todo $p, q, r \in X$ (esto es llamado desigualdad del triangulo).

Una función distancia es además llama métrica. Además, se reserva ese termino para referirnos a la métrica de Riemannian, y se evitará usar el termino para indicar una función distancia para evitar alguna confusión.

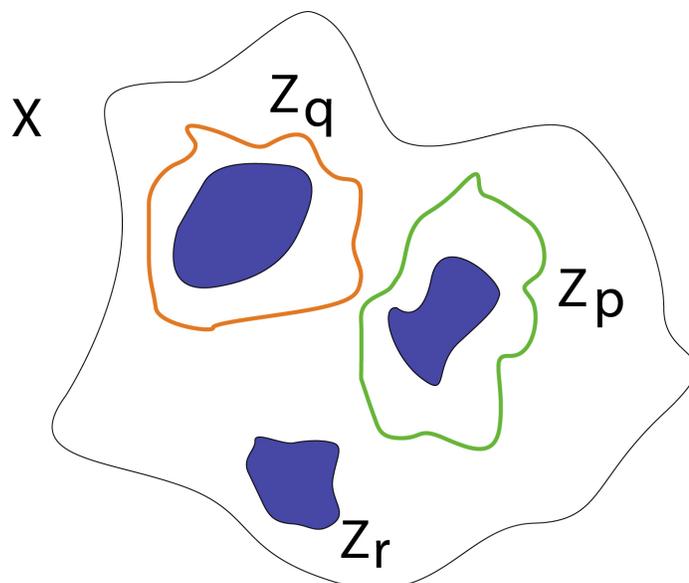


Figura 1.7.7: El grado de homología del grupo da el número de Betti. La clase homológica de un ciclo arbitrario puede expresarse como una combinación lineal de los generadores de los grupos homológicos. $[Z_p]$ y $[Z_q]$ son generadores de $H_1(X)$.

fuelle (Diego Penagos)

Un ejemplo simple para la función distancia incluye p -norma en el espacio vectorial familiar de \mathbb{R}^D , el cual es un espacio topológico con el estándar de topología Euclidiana, con adición de un espacio vectorial). Así, si $x = [x^1, x^2, \dots, x^D]$ y $y = [y^1, y^2, \dots, y^D]$ son puntos en \mathbb{R}^D , entonces $d(x, y) = (|x^1 - y^1| + |x^2 - y^2| + \dots + |x^D - y^D|)^{1/p}$, $p > 0$, define una función de distancia sobre \mathbb{R}^D .

1.7.11. Variedad

Una *variedad* es un espacio topológico que puede ser observado como un espacio Euclidiano en cada punto. Esto es posible, si M es un espacio topológico, y $p \in M$ es un punto sobre el, entonces ahí existe un sector abierto U de p (un conjunto abierto U con $p \in U$), tal que cada uno puede construir homeomorfismos $\phi : U \rightarrow \mathbb{R}^D$ para algún entero positivo D . El valor

mínimo de D para el cual es posible construir tales homeomorfismos es llamado dimensión de la *variedad*.

1.8. ALGORITMO DE BÚSQUEDA GRÁFICA

En una aproximación discreta sobre una ruta de planificación, una gráfica es construida por la discretización del espacio de configuración y el posicionamiento de un vértice / nodo para cada celda discretizada. Los bordes son establecidos sobre la base de acciones disponibles entre vértices vecinos.

Así una gráfica consiste en tres componentes : Un conjunto de vértices $V(G)$, un conjunto de bordes $\varepsilon(G) \subseteq V(G) \times V(G)$, y una función de costo $C_G : \varepsilon(G) \rightarrow \mathbb{R}^+$. Un elemento en $V(G)$ es llamado un vértice o un nodo. Un elemento en $\varepsilon(G)$ está representado por el par ordenado $[a, b] \in \varepsilon(G)$, el cual implica que ahí exista un borde en G conectando $a \in V(G)$ a $b \in V(G)$.

1.8.1. Algoritmo de Dijkstra

El algoritmo de Dijkstra [6] es el mas fundamental en encontrar un costo mínimo (o la distancia mínima) de trayectorias a través de un vértice, $p \in V(G)$, en la gráfica a cualquier otro vértice es accesible desde p . Esto es garantizado para ser óptimo. La intuición detrás del algoritmo de Dijkstra es que empieza desde el inicio de un nodo p , un 'frente de honda' de casi el mismo tamaño que la distancia geodésica (costo del camino mas corto) es propagada a través de la gráfica. La Figura 1.8.2 ilustra el proceso del algoritmo de Dijkstra. Observe que el frente de onda está marcado por los círculos azules vacíos. Esto es un conjunto de nodos $[u \in Q | g(u)$ es infinito].

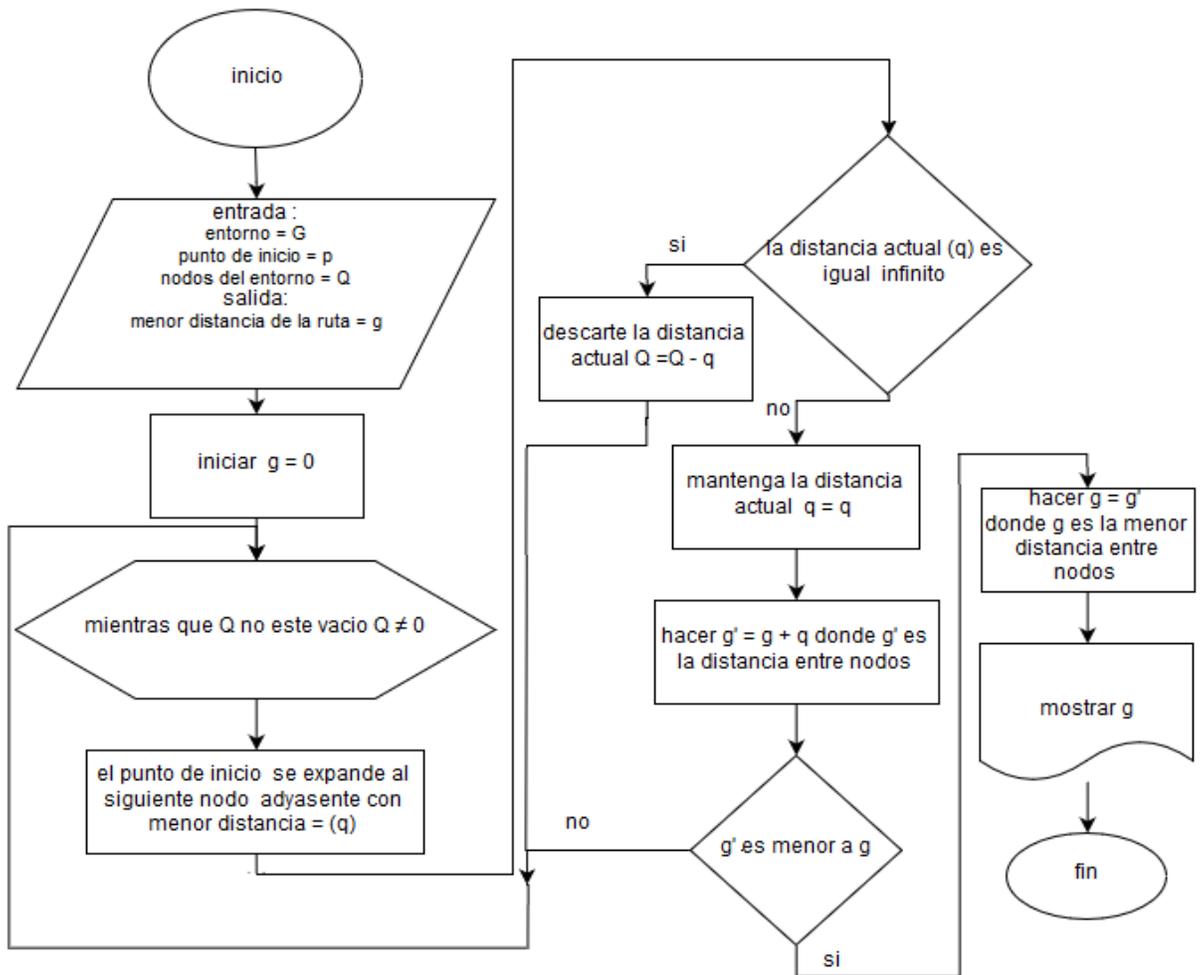


Figura 1.8.1: Representación en diagramas de flujo del algoritmo de Dijkstra.

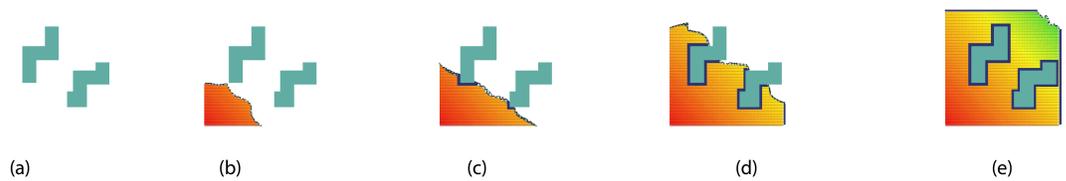


Figura 1.8.2: Ilustración del proceso del algoritmo de Dijkstra. Nodo de inicio en rojo.

fuelle (Diego Penagos)

- (a) *iter* = 0.
- (b) *iter* = 0.25.
- (c) *iter* = 0.5.
- (d) *iter* = 0.75.
- (e) *iter* = 1.

1.8.2. Algoritmo A^*

El algoritmo A^* [8] es, en esencia, muy similar al algoritmo Dijkstra. Mientras que con el algoritmo Dijkstra uno típicamente está interesado en encontrar caminos de costo bajo para cualquier vértice en la gráfica desde un vértice inicial, en A^* se busca un objetivo fijo. Esto permite dirigir la búsqueda de una manera más informada. En lugar de expandir el 'frente de onda' (mencionado anteriormente) uniformemente en todas las direcciones, se expandirá con predilección al vértice seleccionado como el objetivo (Figura 1.8.3). Esta parcialidad está regida por la función heurística. Una función heurística para una gráfica G es una función $h : V(G) \times V(G) \rightarrow \mathbb{R}^+$, tal que $h(v_a, v_b)$ da algunas estimaciones de la mínima distancia (costo total del camino mas corto) entre los vértices v_a y v_b . Una función heurística admisible es aquella función heurística que siempre subestima el valor del costo actual. Esto puede mostrarse en el algoritmo A^* el cual es un heurística admisible que retorna (menos costo) al camino del objetivo. Desde luego, la función constante $h(v_a, v_b) = 0$ es un heurístico admisible, y en este caso puede mostrarse que el algoritmo A^* se convierte en el equivalente del algoritmo Dijkstra. De hecho, la diferencia mas grande entre del algoritmo A^* al algoritmo Dijkstra (junto a otras pocas diferencias estructurales) es que uno selecciona el vértice q a ser expandido como el unico con valor mas bajo de f en lugar de g , donde el valor f es el valor g además de la heurística desde el vértice hasta el objetivo.

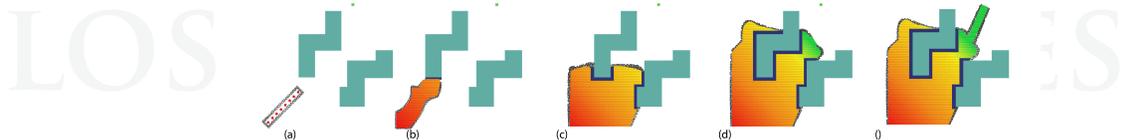


Figura 1.8.4: Ilustración del proceso del algoritmo A^* . El conjunto abierto esta marcado por los círculos azules vacíos.

fuentes (Diego Penagos)

- (a) $iter = 0$.
- (b) $iter = 0.25$.
- (c) $iter = 0.5$.
- (d) $iter = 0.75$.
- (e) $iter = 1$.

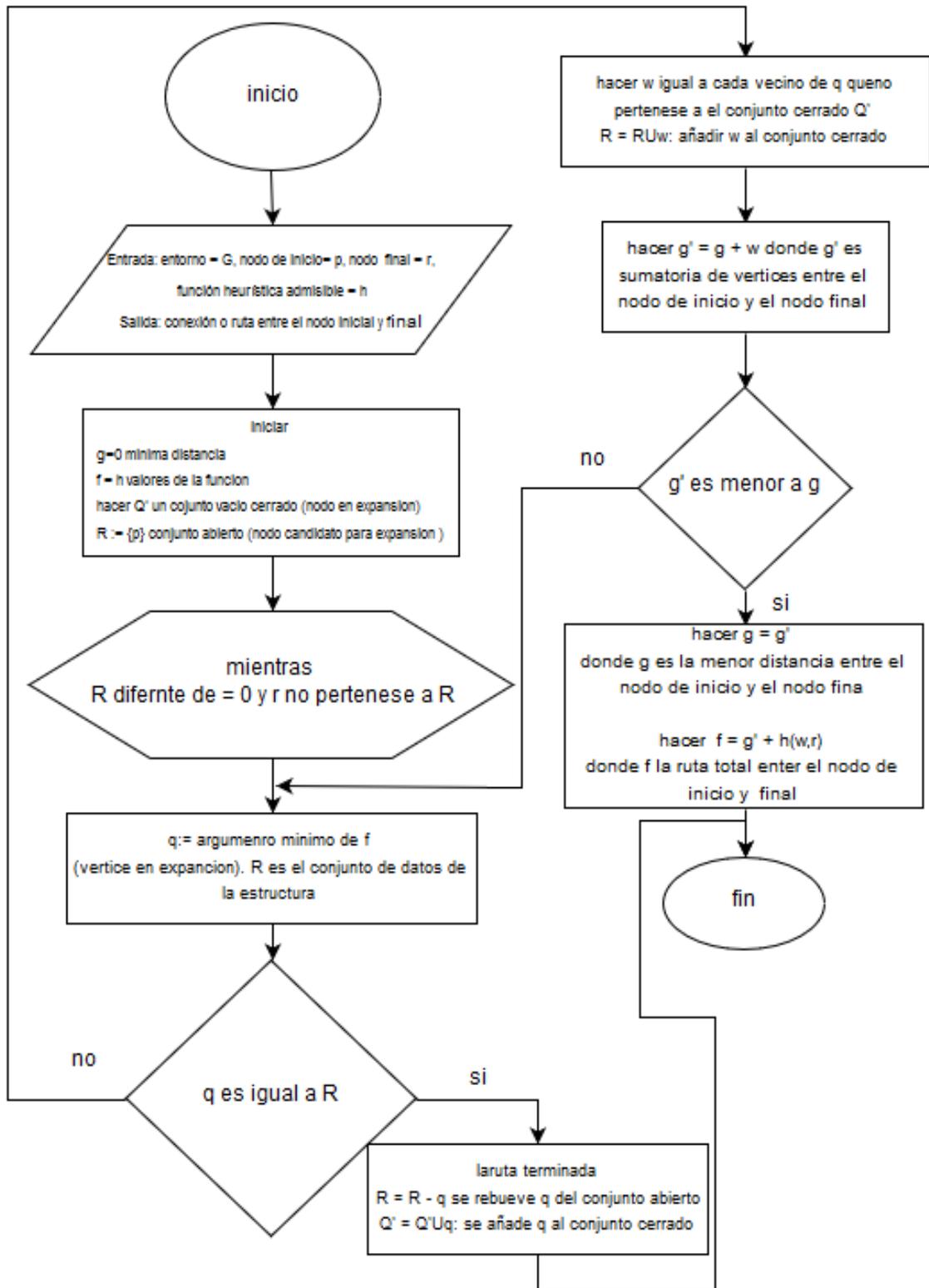


Figura 1.8.3: Representación en diagramas de flujo del algoritmo de A^* .

2. IMPLEMENTACIÓN

2.1. CREACIÓN DEL ENTORNO

El objetivo de este paso consiste en que se pueda elegir si se introduce el entorno de trabajo por medio de una imagen en blanco y negro, donde los obstáculos tienen que estar delimitados por una línea de color blanco y el resto del entorno de color negro, esto para poderlos asociar a sus respectivos conjuntos o elegir que el algoritmo le genere un entorno de un tamaño determinado creando obstáculos aleatorios.

De acuerdo a las definiciones vistas anteriormente, se ve a un grupo de obstáculos y el espacio resultante como una topología donde $\mathcal{T} = (X, \emptyset)$ siendo \mathcal{T} el entorno X los obstáculos y \emptyset el resto del entorno por donde se crearán los caminos a recorrer, puesto que cumple con la Definición 1.1, llegando al punto de tratarse de forma general como el conjunto topológico.

Para el diseño del entorno se mantuvieron cuatro características principales, el número de obstáculos, la cantidad de formas que se pueden tener, la posición en la que se encuentra cada uno en el entorno y su tamaño. Cada una de estas características se generó a través de un número aleatorio por medio de la función *rand*, sin embargo para el tamaño y para la cantidad de obstáculos se proporcionó un valor mínimo y uno máximo, de esta forma se controla la creación de obstáculos muy pequeños, muy grandes o casos en los que hayan tantos obstáculos que ocupen todo el entorno. El tamaño por defecto varía entre el 5% y el 10% de la longitud del entorno, mientras que varían entre 10 y 20 la cantidad de obstáculos.

Se consideraron cuatro formas posibles para cada obstáculo: cuadrados, círculos, triángulos y rectángulos, donde se ve que estas cuatro figuras hacen parte del conjunto x donde $x \in X$: teniendo esto en cuenta se cumple con la definición 1.1 donde los elementos que conforman el subconjunto x se unen e interceptan entre sí con sus puntos límites definidos.

Como se puede ver en la siguiente imagen se muestra uno de los posibles resultados de la creación del entorno.

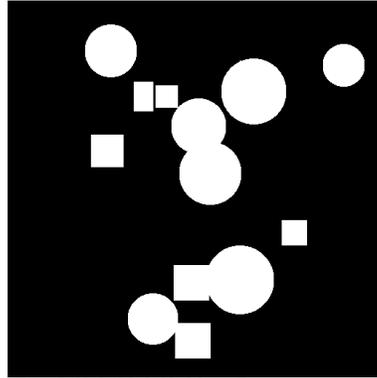


Figura 2.1: Entorno aleatorio creado por el algoritmo 1.

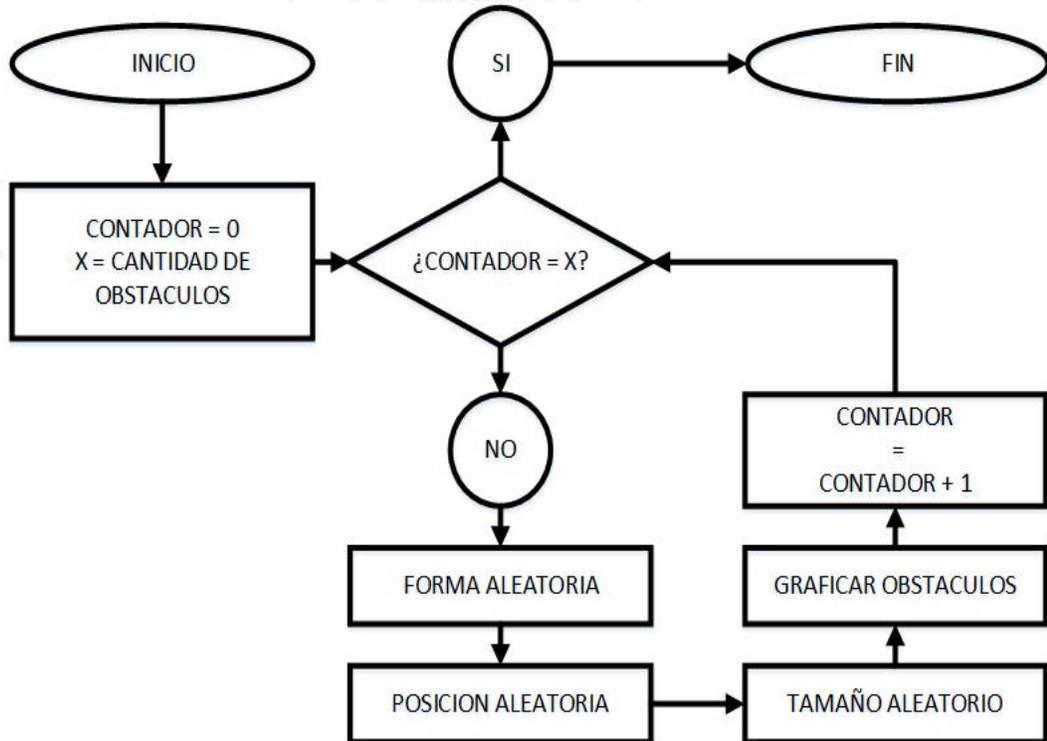


Figura 2.2: Diagrama de flujo de Entorno aleatorio.

2.2. DISCRETIZACIÓN DEL ENTORNO

Se procede a discretizar \mathcal{T} siendo un conjunto indiscreto, separando de este la sección a utilizar, siendo \emptyset el subconjunto de espacio en donde se aplicaran las teselaciones, se procede a separar los obstáculos del camino a recorrer con el siguiente procedimiento, $X \cap \mathcal{T} = \emptyset$ teniendo a \emptyset separado de X lo que se tiene como obstáculos; se procede a generar subconjuntos discretos de \emptyset , cabe antes aclarar que teniendo separados X de \emptyset se tomar X y separarlo por subconjuntos de x_n donde $x_n \in X$, siendo cada subconjunto x_n un obstáculo se procede a darle a cada uno una numeración, para después hacer los cálculos de la cantidad de las teselaciones a aplicar dependiendo del número de obstáculos que se tienen dentro del entorno.

En el caso que se llegue a realizar una discretización del entorno, con sus respectivas uniones de obstáculos e intersecciones de las teselaciones sobre un espacio abierto o disponible, se puede considerar una topología cofinita; ya que con esto se asegura que el entorno se logre discretizar. Para ello se genera la cantidad de teselaciones necesarias teniendo en cuenta la cantidad de obstáculos con el espacio ocupado por los mismos en el entorno y la expansión del frente de onda de cada una de ellas, siempre y cuando sea válida (se considera una teselación válida si su punto de origen no se encuentra contenida en x). La primera característica a tener en cuenta es la cantidad de teselaciones a utilizar.

Al trabajar con un conjunto discreto, se puede aplicar la topología euclidiana, incluyendo aquí mismo los espacios métricos, por el momento cumpliendo con las reglas de la topología euclidiana se toma el conjunto \emptyset y se trabaja en \mathbb{R}^2 lo que se quiere a continuación es comparar nuestro conjunto X con el conjunto \emptyset y se puede calcular la media y la covarianza, luego por medio de la media y la covarianza se encuentra el valor mínimo necesario. El número encontrado según el cálculo posee una raíz cuadrada entera por lo que se esta trabajando en un entorno euclidiano y métrico, con esto se tiene una cantidad apropiada de teselaciones a realizar teniendo en cuenta que si $X > \emptyset$ el número de teselaciones aumentará pero si el área que ocupa $X <$ al área que ocupa \emptyset el número de teselaciones disminuirá, no todas las teselaciones son válidas.

En el algoritmo 1 se encuentran los pasos realizados mediante un pseudocódigo.

```

I = Imagen de entrada
U = Vector para definir el area de cada obstaculo
G = Area total ocupada por los obstaculos
E = Numero total de pixeles de la imagen del entorno
D = Cantidad de obstaculos
X = Numero de teselaciones
//Conteo y etiquetado de obstaculos a traves de la funcion bwlabel(I)
D = bwlabel(I)
FOR{i = 1 to E}
\IF {I(i) = E}
  {U(i) = U(i) + 1}
\ENDIF
\ENDFOR
// Calculo de la media y la covarianza a partir de U y D
{X = (covarianza + media + D)/3}

```

Algoritmo 1 pseudocodigo Numero optimo de teselaciones

2.3. EXPANSIÓN Y LIMITACIÓN DE LAS TESELACIONES

Una de las partes importantes de la discretización de entornos es la forma de hacer el muestreo de los espacios sin obstáculos anterior mente llamado conjunto básico \emptyset , por lo que se quiere hacer una expansión de varios frentes de ondas que se van a expandir hasta encontrar un obstáculo siendo este parte de X o hasta encontrar otro frente de onda, en este caso se mostrarán las expansiones hechas por frentes de onda con polígonos regulares por lo que estos ofrecen facilidad a la momento de que estos se expandan de forma uniforme como son: círculo cuadrado y rombo; se realizarán de forma ordenada para poder ver con más detalle esta expansión, en las siguientes imágenes se muestra el proceso de la expansión de las teselaciones en el entorno, para ver más clara mente lo que se quiere es crear sub espacios dentro del conjuntos \emptyset , este está formado por área la cual se discretizar gracia a la topología euclidiana, así mismo cada uno de los sub conjuntos creados dentro del conjunto básico \emptyset tienen que cumplir con las reglas de la topología dándole a cada uno puntos limites como se explicó previamente.

Teselaciones a partir de Cuadrados:

Se pensó aplicar los frentes de onda por cuadrados ya que es la forma mas uniforme en un plano cartesiano; por lo que en los cuatro cuadrantes se tiene la misma área ocupada , al tener dos simplisiales expandiendose por cada cuadrante

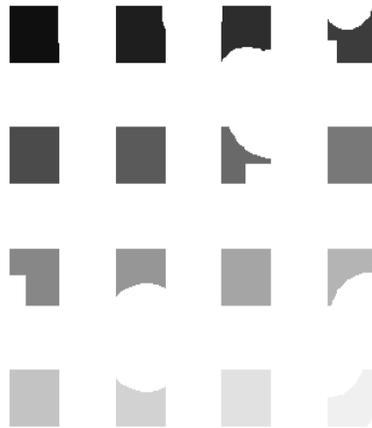


Figura 2.3: Diagrama de Teselaciones a partir de Cuadrados iter = 0.13.



Figura 2.4: Diagrama de Teselaciones a partir de Cuadrados iter = 0.33.

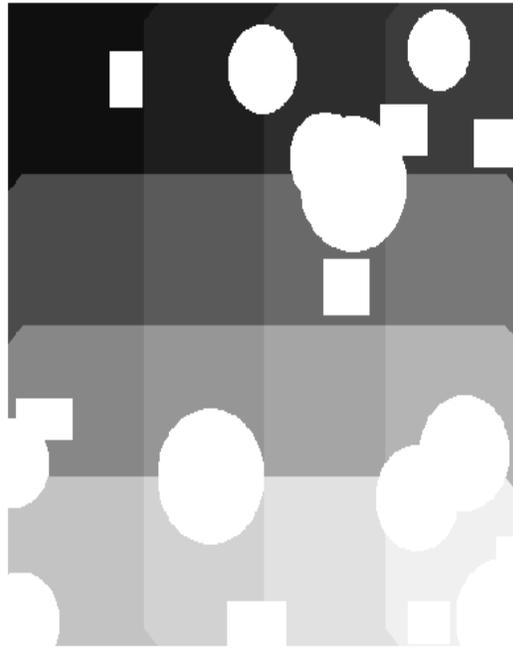


Figura 2.5: Diagrama de Teselaciones a partir de Cuadrados iter = 0.66.

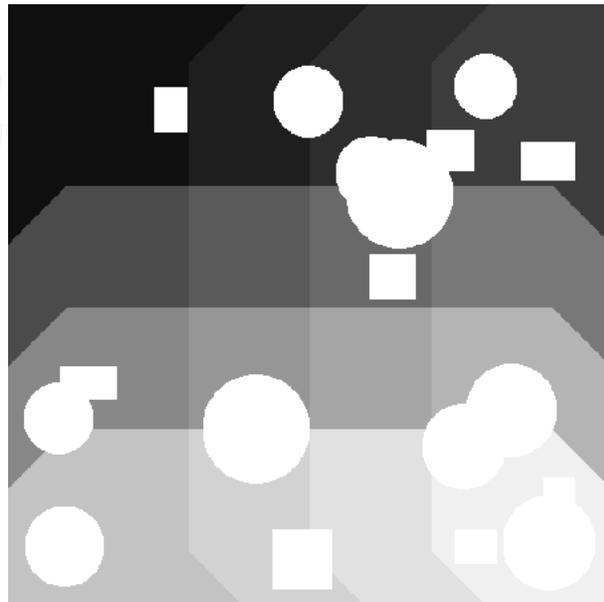


Figura 2.6: Diagrama de Teselaciones a partir de Cuadrados iter = 1.

Teselaciones de referencia:

Con las siguientes teselaciones se quiere obtener un punto de comparacion mas amplio a parte de los tes anteriores mencionados

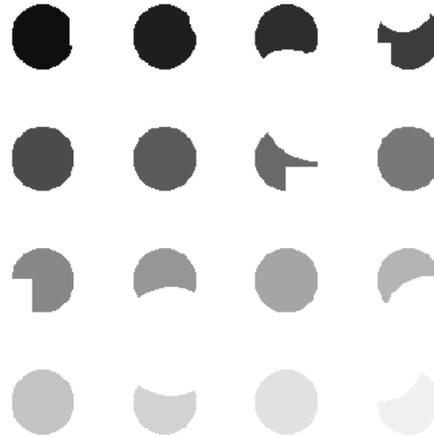


Figura 2.7: Diagrama de Teselaciones a partir de Círculos iter = 0.13.

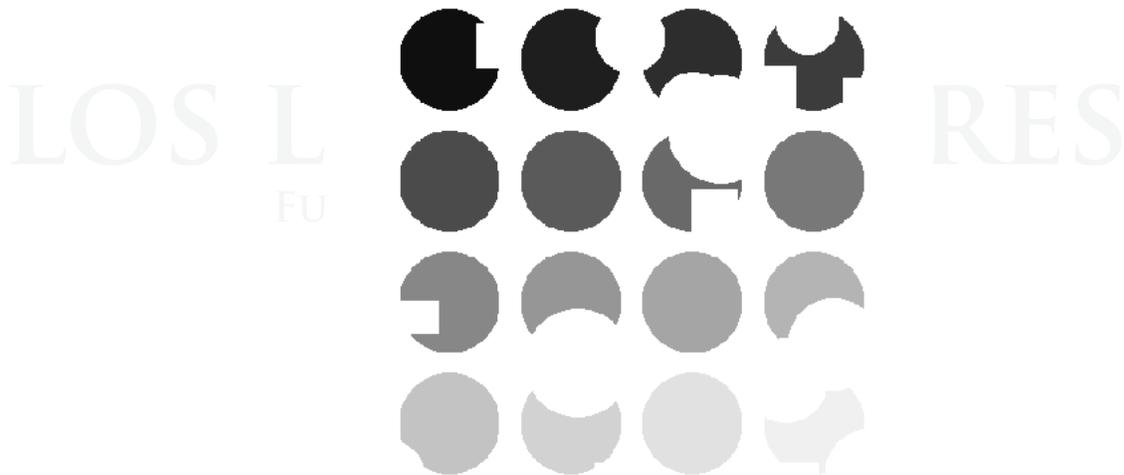


Figura 2.8: Diagrama de Teselaciones a partir de Círculos iter = 0.33.

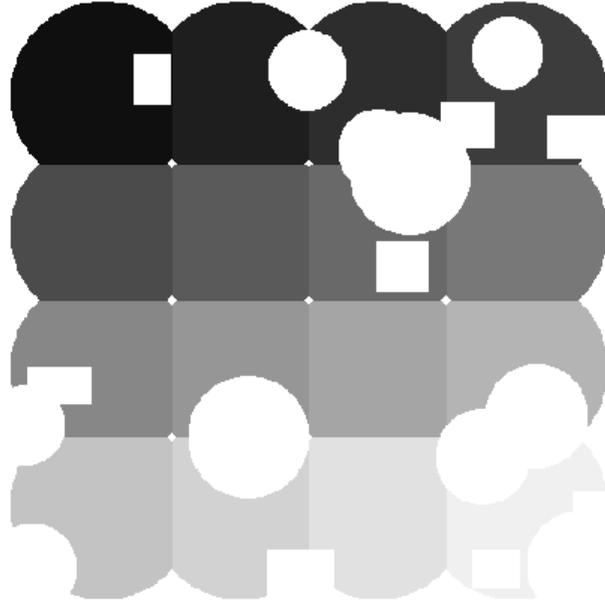


Figura 2.9: Diagrama de Teselaciones a partir de Círculos iter = 0.66.

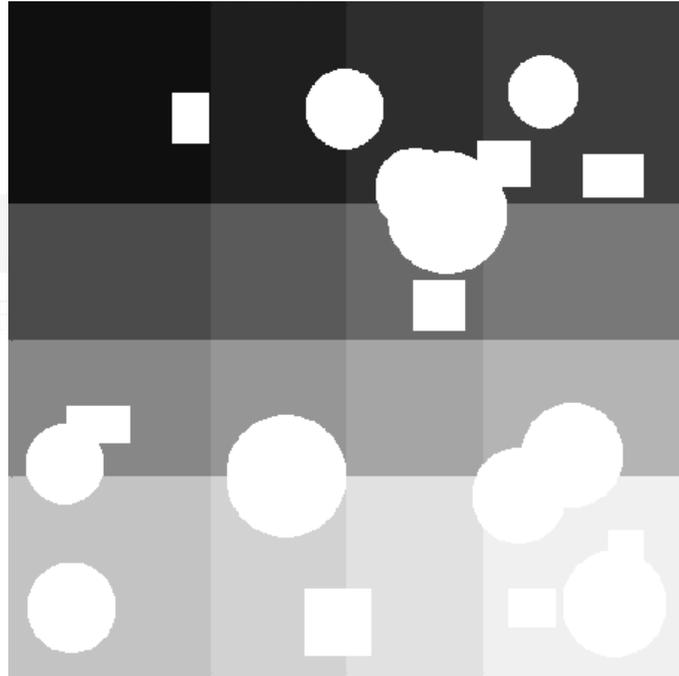


Figura 2.10: Diagrama de Teselaciones a partir de Círculos iter = 1.

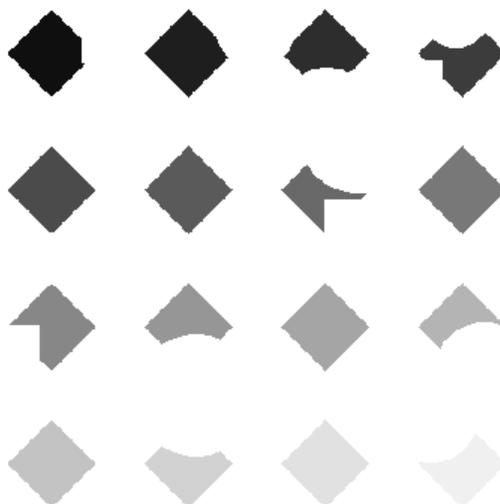


Figura 2.11: Diagrama de Teselaciones a partir de rombos iter = 0.13.



Figura 2.12: Diagrama de Teselaciones a partir de rombos iter = 0.33.

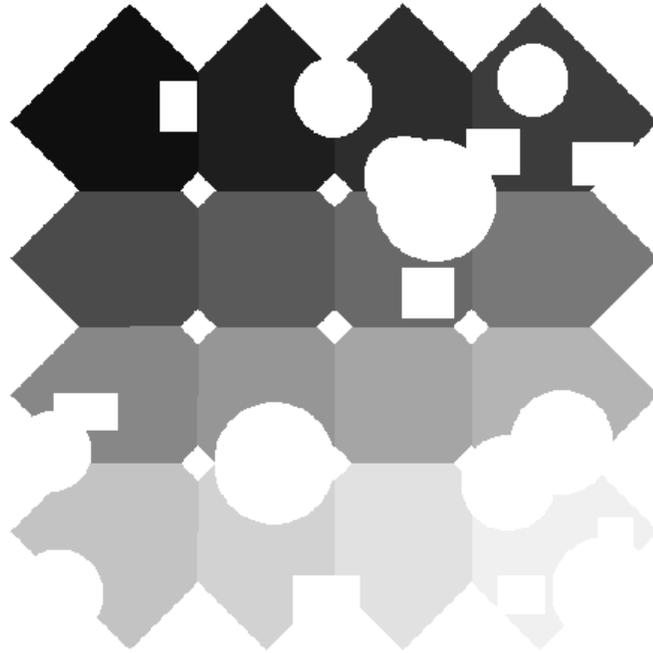


Figura 2.13: Diagrama de Teselaciones a partir de rombos iter = 0.66.

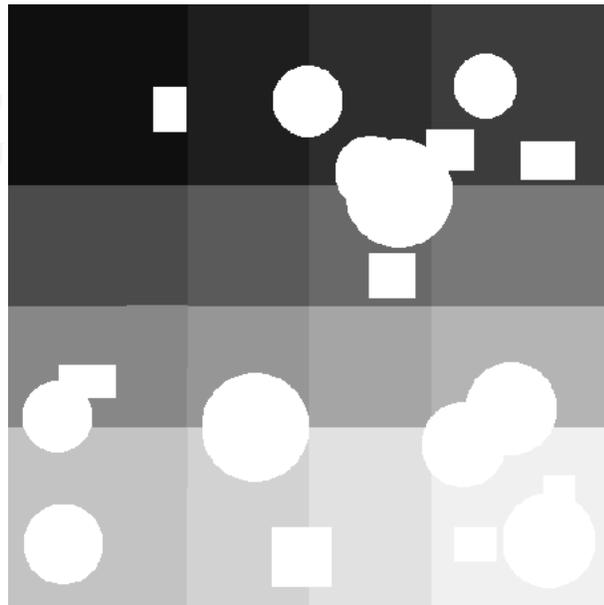


Figura 2.14: Diagrama de Teselaciones a partir de rombos iter = 1.

2.4. UBICACIÓN DE LAS TESELACIONES

Ya teniendo el número óptimo de las teselaciones a aplicar para poder discretizar el entorno se procede a ubicarlas en este, se comenzó haciendo este proceso de orden aleatorio para obtener el resultado que se ve en la figura 2.15, por lo que se obtuvieron varios polígonos irregulares después de esto se prosiguió a darles un orden uniforme a la posición de los centros de las teselaciones lo que se vio como resultado una mayor cantidad de polígonos regulares como se ve en la figura 2.16, después de esto se quiso ver el resultado de las teselaciones con respecto a las áreas que ocupaban en el entorno, al comparar los resultados entre las áreas de las teselaciones aleatorias y las ubicadas equidistantes se encuentra que al tener las teselaciones ordenadas las áreas de estas son más uniformes estando entre un promedio del área total de las teselaciones, por lo que más adelante esto ayudará a hacer más eficiente la expansión de los frentes de onda, gracias a esto todas van a gastar el mismo ciclo de maquina expandiéndose, ya que se esta trabajando en un entorno de \mathbb{R}^2 , al realizar la expansión por frentes de onda con forma de polígonos regulares, esta expansión debe cumplir con la topología euclidiana como anteriormente se menciona, esta expansión se hará de forma sucesiva y considerando los valores adyacentes al punto anterior.

El frente de onda se detendrá una vez encuentre un obstáculo o un frente de onda generado por otra teselación. Cada teselación generará el frente de onda con un color específico (en escala de grises); el objetivo de graficar cada frente de onda de distinto color es poder diferenciar uno del otro.

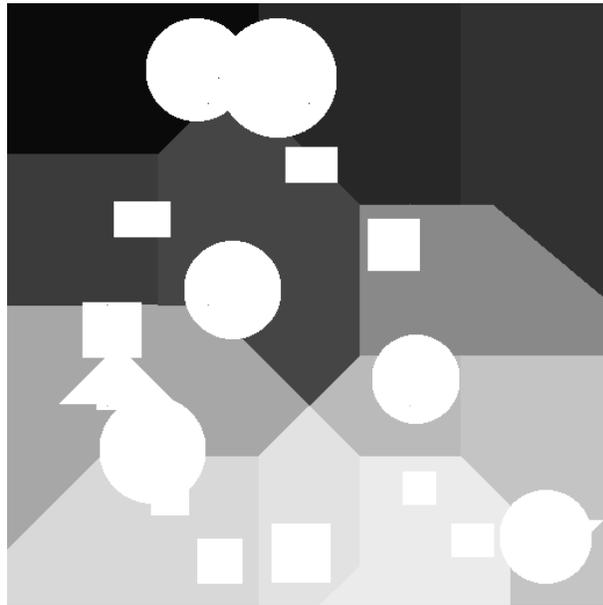


Figura 2.15: Entorno con teselaciones aleatorias .

X = Numero de teselaciones

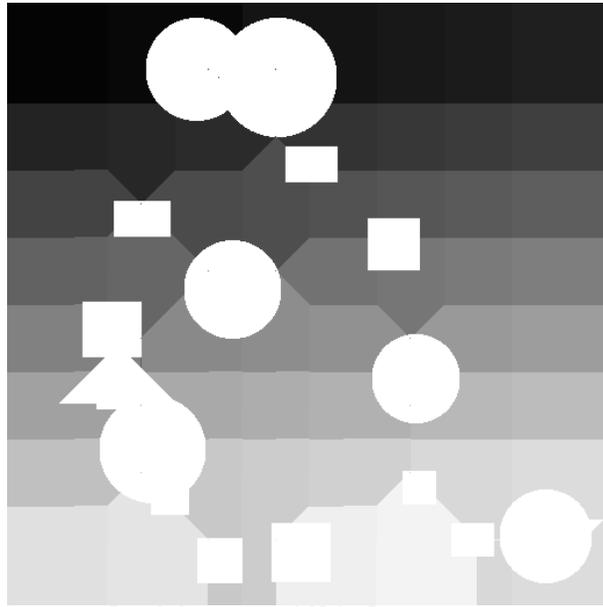


Figura 2.16: Entorno con teselaciones ordenadas.

```

Y = Posicion de cada teselacion
G(1,X) = Tonalidad de cada teselacion (Escala de grises)
Z = Numero de puntos por expandir
I = Imagen de entrada
V = Numero de puntos expandidos
W = Contador actual de teselaciones
J = Imagen de salida

    WHILE { Z != V }
    WHILE { X != W }
//Expandir frente de onda en una posicion
    IF { I(Y) = 0 AND J(Y) = 0 }
        { J(Y) = G(1,W) }
        { V = V + 1; }
    ENDIF
        { W = W + 1 }
    ENDWHILE
ENDWHILE

```

Algoritmo 2 pseudocodigo Teselaciones

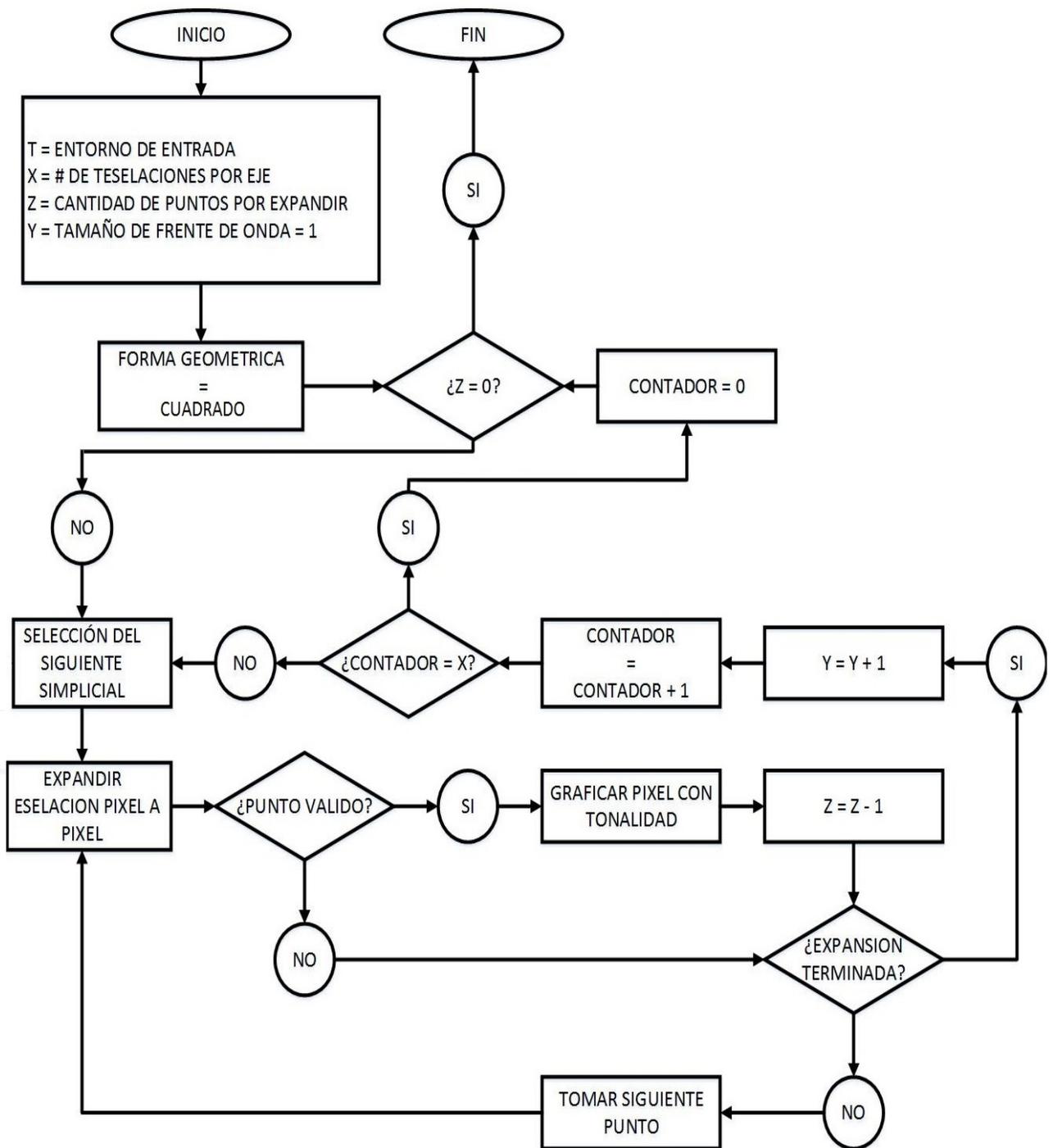


Figura 2.17: Diagrama de flujo de expansión de teselaciones .

2.5. COMPROBACIÓN DE UBICACIÓN DE LAS TESELACIONES

Para el comienzo de la ubicación de las teselaciones se inició haciendo un análisis por medio de su ubicación de forma aleatoria, donde se obtuvo como resultado varios polígonos irregulares como se puede ver en la figura 2.15, después se planteó colocar los centros de las teselaciones con un orden matricial en filas y columnas como se ve en la figura 2.16, por lo que se obtuvo una mayor cantidad de polígonos regulares, los que al momento de hacer el cálculo de sus centros de masa se encuentra que es más rápido procesar las imágenes con teselaciones ordenadas a comparación del tiempo que se demora hacer el cálculo de los centros de masa con teselaciones aleatorias, con rápido se quiere hacer referencia al menor número de ciclos de máquina esto se verificó con 10 imágenes diferentes donde a cada una se le aplicaron las teselaciones aleatorias y uniformes dando una diferencia promedio de ± 2 segundos entre las dos.

Para realizar la comprobación de la forma más óptima para colocar los centros de las teselaciones en el entorno, se quiso comprobar que la suma de los ángulos de las esquinas entre 2 o más teselaciones sea 360° dentro del entorno, aunque en algunos casos los ángulos que dan no cumplen esta condición por lo que estos ángulos se suman hasta completar 360° cuando esta condición se cumple este ángulo se hace 0 y en el caso que el entorno este mal discretizado se tiene un residuo, el cual se divide entre 360° y se multiplica por 100 para tener el porcentaje de cubrimiento del entorno en donde se puede notar que los teselados colocados de forma aleatoria tenían un rango de 0.5% a un 0.8 % del entorno que no fue discretizado a comparación de cuando se colocaron los centros de las teselaciones de forma equidistante “filas y columnas” en donde se obtuvo entre 0.1% y 0.5 %.

Después de esto se verificó este resultado con los demás entornos donde este rango variaba, al observar los entornos por separado se veía que mientras más circunferencias el entorno tenía mayor era el porcentaje de error, por lo que se pasó a otro método para verificar si el entorno quedaba bien discretizado. Por lo que se decidió hacer un programa basado en el algoritmo figura 2.18 ya que el entorno para este punto ya se encuentra dentro de la topología euclidiana y se le han dado parámetros métricos se puede comparar los puntos que se encuentran dentro y fuera de las teselaciones lo que se hace es comparar la matriz de lectura del entorno con la matriz de salida de las teselaciones para ver qué área no ha sido cubierta por las teselaciones o si un obstáculo fue cubierto por las teselaciones por lo que no se quiere que esto se cumpla solo se quiere tener discretizado el entorno por donde se quiere circular sin perder las posibles rutas a recorrer

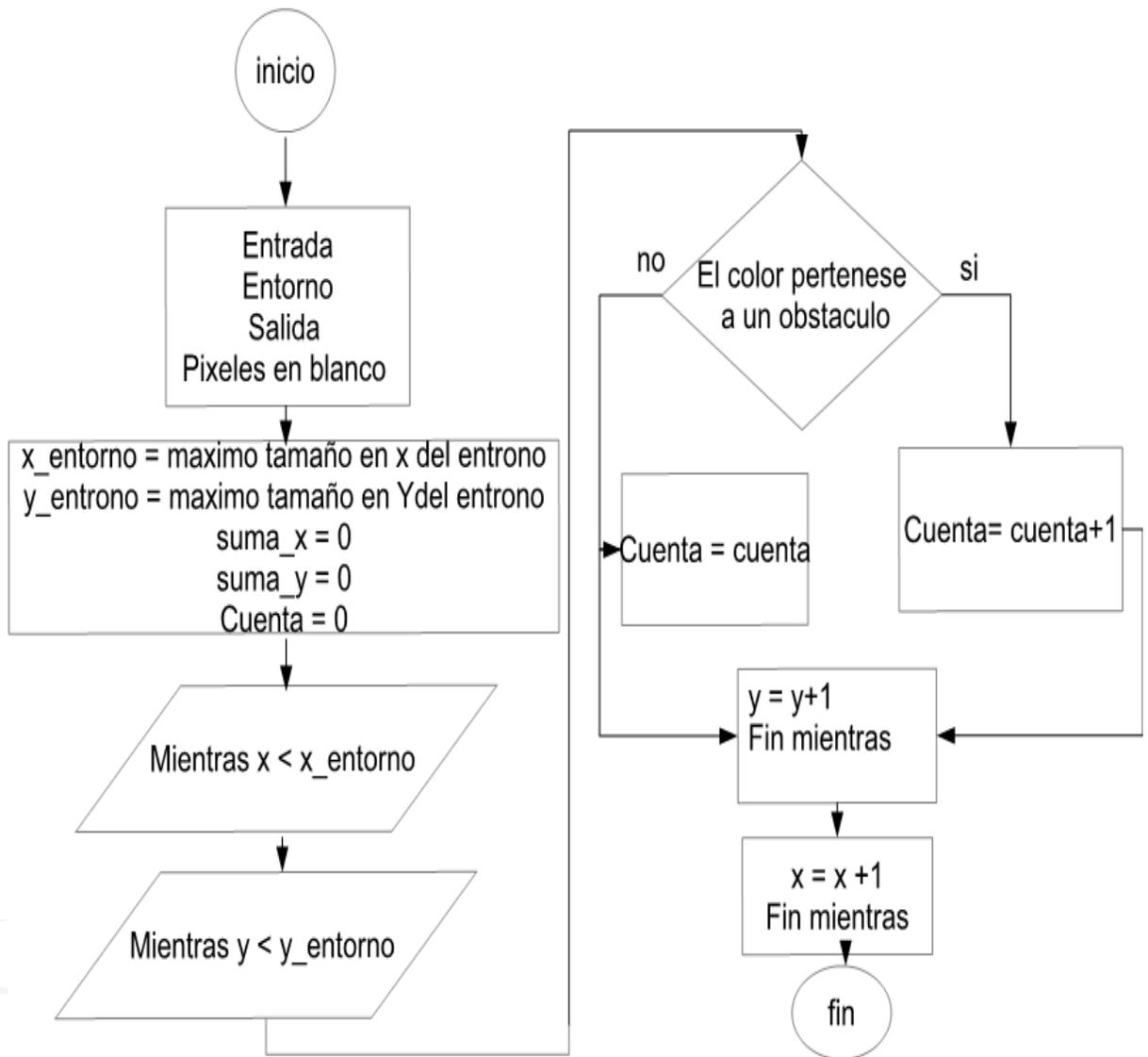


Figura 2.18:Diagrama de área no cubierta.

para finalizar se toman las teselaciones de forma ordenada y se desplazaron las filas y las columnas para ver si esto alteraba en algo la discretización del entorno como se ve en las figuras 2.19 y 2.20

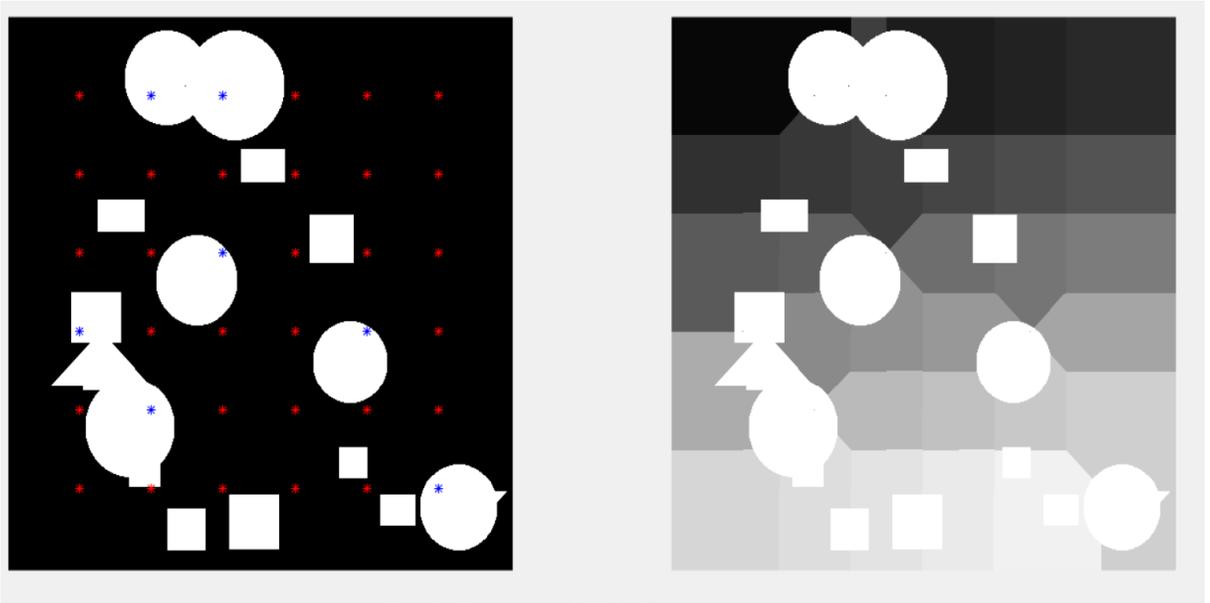


Figura 2.19:Entorno con teselaciones ordenadas .

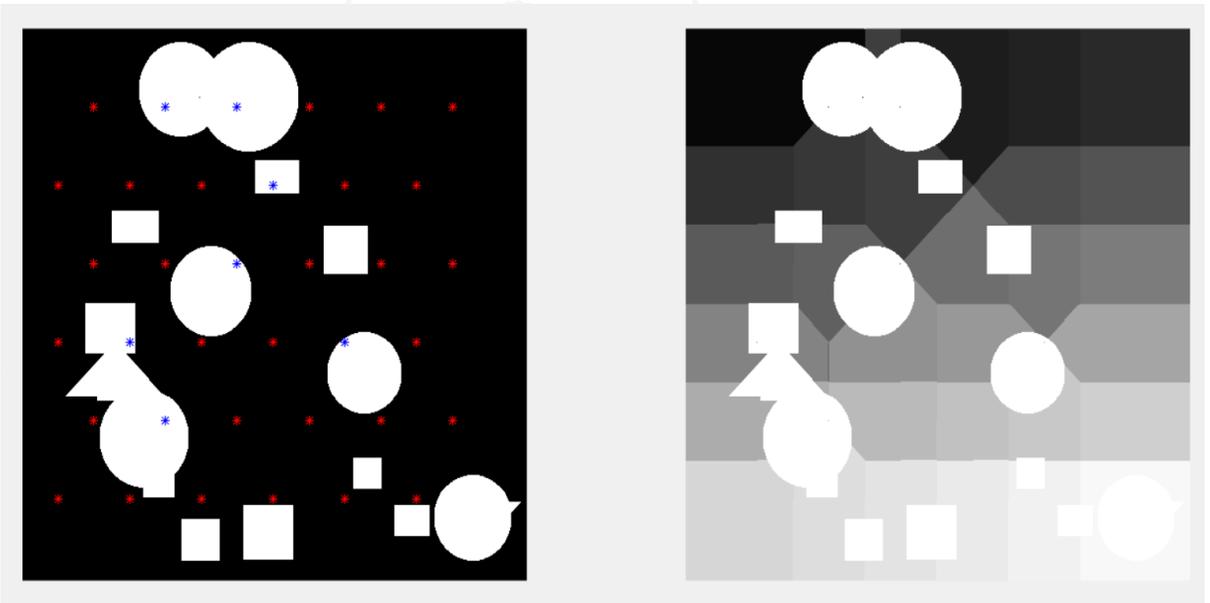


Figura 2.20:Entorno con teselaciones ordenadas con filas desplazadas .

2.6. CALCULO DE CENTROS DE MASA

Por último se toma cada teselación por separado, para realizar el cálculo del centro de masa de cada teselación, teniendo en cuenta que se esta trabajando en \mathbb{R}^2 ya se tiene el entorno discretizado e implementado en una matriz por lo que se selecciona el área de cada teselación para eliminar los puntos límites de cada teselación y de este modo con un producto punto se puede hallar el centro de masa para una teselación este proceso se repite por cada teselación. Teniendo como base la figura 2.10, se procede a obtener los centros de masa de cada área conformada por las teselaciones como se ve en la figura 2.21

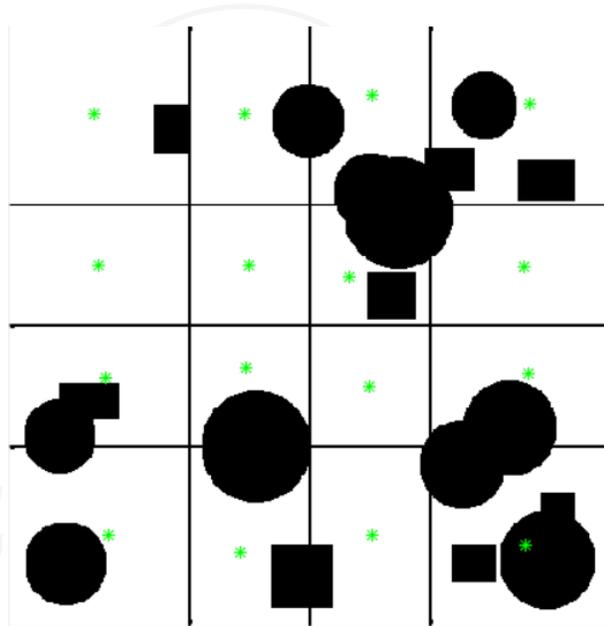


Figura 2.21: centros de masa de teselaciones ordenadas.

```

X = Numero de teselaciones
J = Imagen de entrada
F = Cantidad de filas de J
C = Cantidad de columnas de J
G(1,X) = Tonalidad de cada teselacion (Escala de grises)
H = Imagen de transicion
W = Contador actual de teselaciones
S = Matriz de unos para el calculo del centroide
K = Imagen de salida
C(x,y) = Posicion del centroide de cada teselacion

WHILE { W != X }
  FOR{ i = 1 to F }
    FOR{ j = 1 to C }
      IF { J(Y) = G(1,W) }
        { H(i,j)=255 }
      END IF
    END FOR
  END FOR
  FOR{ i = 1 to F }
    { M1 = i x (H X S)+M1 }
    { M2 = (H X S)+M2 }
  END FOR
  FOR{ j = 1 to C }
    { M3 = j x (H X S)+M3 }
    { M4 = (H X S)+M4 }
  END FOR
  { C(x) = M1/M2 }
  { C(y) = M3/M4 }
  { W = W + 1 }
END WHILE

```

Algoritmo 3 pseudocodigo centros de masa de Teselaciones

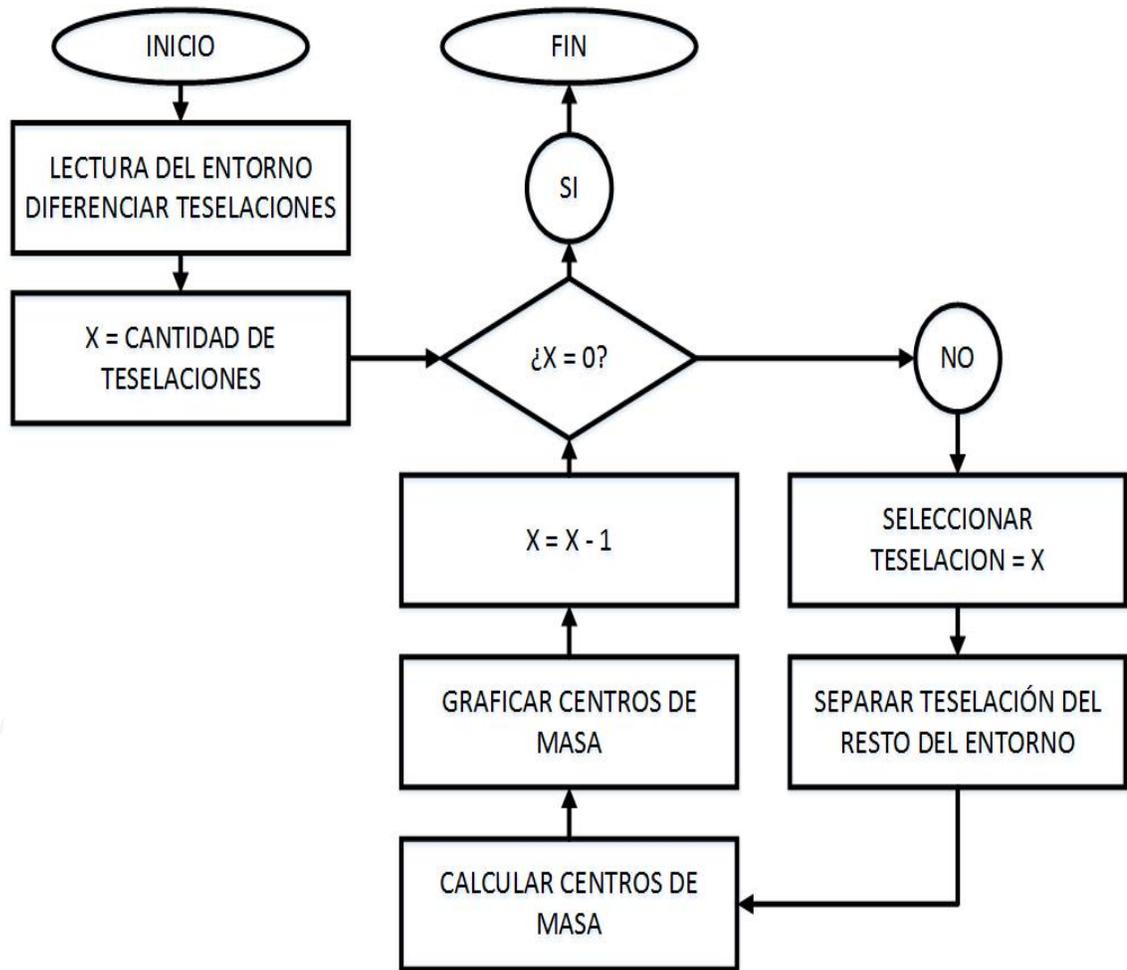


Figura 2.22: Diagrama de flujo de centros de masa de teselaciones .

2.7. RESULTADOS

Con el procedimiento anterior se toman los datos de 10 entornos diferentes, por lo que esto están basados en imágenes de 520 X 520 lo que indica que estas están formadas por un total de 270400 píxeles los cuales fueron analizados y discretizados el proceso se tomó teniendo en cuenta los obstáculos que estaban representados de un color negro al comienzo del análisis, después este dato se guardó para su comparación con la discretización final del entorno, con esto puede tener un promedio de la cantidad de píxeles que no fueron discretizados de forma correcta obteniendo los siguientes datos, entre ellos están los datos de 10 entornos con teselaciones colocadas de forma aleatoria y las teselaciones colocadas de forma equidistante, entre ellas de forma matricial teniendo filas y columnas, y la expansión de sus frentes de onda con polígonos regulares por lo que primero se obtuvo la siguiente tabla donde se toman los datos de los teselados con ubicación aleatoria y frentes de onda cuadrado, círculo y rombo

Entorno	número de píxeles no leídos		
	círculo	rombo	cuadrado
1	136	29	30
2	82	55	58
3	109	3	5
4	55	57	56
5	28	4	57
6	217	56	29
7	190	55	59
8	56	3	4
9	163	0	60
10	109	28	31

Tabla 2.1: número de píxeles no leídos con teselaciones aleatorias.

Entorno	porcentaje del entorno no leído		
	círculo	rombo	cuadrado
1	0.05%	0.01%	0.01%
2	0.03%	0.02%	0.02%
3	0.04%	0.001%	0.001%
4	0.02%	0.02%	0.02%
5	0.01%	0.001%	0.02%
6	0.08%	0.02%	0.01%
7	0.07%	0.02%	0.02%
8	0.02%	0.001%	0.001%
9	0.06%	0%	0.02%
10	0.04%	0.01%	0.01%
Total	0.042%	0.0103%	0.0132%

Tabla 2.2: área no cubierta por teselaciones aleatorias.

Para finalizar se hizo el mismo procedimiento con las teselaciones ubicadas de forma ordenada con lo que se obtuvo un error del 0% como se ve en la siguiente tabla comparando los resultados de los datos de las teselaciones ordenadas y las colocadas de forma aleatoria.

frente de onda	orden	aleatorio
Cuadrado	0%	0.0132 %
círculo	0%	0.042 %
Rombos	0%	0.0103 %

Tabla 2.3: área no cubierta por teselaciones.

a continuación se ve las tablas de los tiempo y el número de ciclos de los algoritmos encargados de generar ejecución de los frentes de onda dependiendo de su forma de expansión, ya teniendo pre establecido la ubicación de las teselaciones en cada uno de los entornos.

Código	Llamadas	Tiempo total	tiempo %
Figura 1.1	1	0.134 s	12.8 %
Figura 1.2	1	0.057 s	5.5 %
if (frontera)	176384	0.036 s	3.4 %
end (if)	270400	0.033 s	3.2 %
if (espacio)	270400	0.029 s	2.8 %
Otros comandos		0.755 s	72.3 %
Total		1.044 s	100 %

Tabla 2.4: tiempos de ejecución para cuadrados.

Código	Llamadas	Tiempo total	tiempo %
Figura 1.1	54091600	5.680 s	11.1 %
Figura 1.2	54091600	5.508 s	10.7 %
if (frontera)	54091600	6.225 s	12.2%
end (if)	54091600	5.501 s	10.7 %
if (espacio)	54091600	5.480 s	10.7 %
Otros comandos	54091600	22.978 s	44.7 %
Total		51.402 s	100 %

Tabla 2.5: tiempos de ejecución para círculos.

Código	Llamadas	Tiempo total	tiempo %
Figura 1.1	1	0.150 s	10.3 %
Figura 1.2	1	0.055 s	3.8 %
if (frontera)	347760	0.048 s	3.3 %
end (if)	347760	0.047 s	3.2 %
if (espacio)	347760	0.047 s	3.2 %
Otros comandos		1.116 s	76.3 %
Total		1.463 s	100 %

Tabla 2.6: tiempos de ejecución para rombos.

3. CONCLUSIONES

Teniendo un conocimiento básico de la topología puede ver que estos algoritmos pueden ser aplicados a cualquier entorno o espacio topológico, desde que este cumpla con las definiciones básicas de la topología y teniendo en cuenta sus límite de dos dimensiones.

Cuando se quiso hacer la comprobación del recubrimiento de las teselaciones al entorno por medio de la sumatoria de ángulos, esta salía errónea por lo que en algunas esquinas se encontraron circunferencias, que al momento de hacer el cálculo de los ángulos por medio de vectores de una matriz no se tomaba un punto exacto para hacer el cálculo.

La mejor forma de ubicar los centros de las teselaciones es de forma equidistante entre ellas, entre filas y columnas por lo que al momento de generar el frente de onda de cada una de las teselaciones sus áreas se mantienen en un promedio, esto genera que los tiempos de expansión sean iguales y terminen al mismo tiempo, al momento de encontrar el centro de masa de cada teselación se mantenga una uniformidad de distancia entre ellas, con esto se hace más rápido el cálculo de una ruta entre dos puntos.

Teniendo en cuenta las tablas de resultados se ve que una de las formas menos óptimas de hacer una expansión del frente de onda para generar la teselación es de forma circular, esto es debido al hecho de que el entorno está basado en cuadrados "píxeles" por lo que al momento de querer tener una circunferencia se hace necesario tener la mayor cantidad de píxeles posibles lo que genera un gasto computacional más alto a comparación de las otras formas de expansión de los frentes de onda.

Haciendo la comparación entre las tres formas de expansión de los frentes de onda se ve que la expansión hecha con la forma de cuadrado es la más eficiente para este caso por lo que alcanza a cubrir todos los píxeles como se puede ver en la tabla 2.4 sin necesidad de hacer una repetición de ciclo, de resto se ve que la cantidad de veces que se repiten los ciclos de lectura de los píxeles son mayores a la cantidad de píxeles que se encuentran en la imagen que representa un entorno.

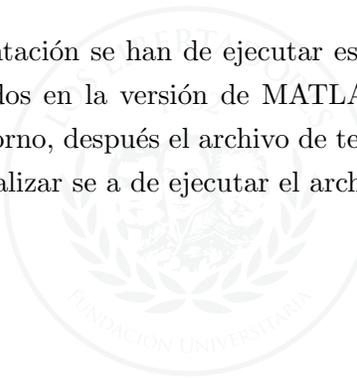
La mejor forma de ubicar los centros de las teselaciones es de forma equidistante entre ellas, entre filas y columnas por lo que al momento de generar el frente de onda de cada una de las teselaciones sus áreas se mantienen en un promedio, esto genera que los tiempos de expansión sean iguales y terminen al mismo tiempo, al momento de encontrar el centro de masa de cada teselación se mantenga una uniformidad de distancia entre ellas, con esto se hace más rápido el cálculo de una ruta entre dos puntos.

4. ANEXOS

En el CD en la carpeta Algoritmos se encuentran los algoritmos implementados en MATLAB con su extensión .tex se encuentra:

Centroides, Entorno, teselaciones_circulos, teselaciones_cuadrados, Teselaciones_rombos

Para una buena implementación se han de ejecutar estos archivos en la misma carpeta y es recomendable ser ejecutados en la versión de MATLAB 2013 o superior, primero se ha de ejecutar el archivo de Entorno, después el archivo de teselaciones_” ” dependiendo del tipo de frente de ondas y para finalizar se a de ejecutar el archivo de Centroides



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

BIBLIOGRAFÍA

- 1 TOPOLOGICAL AND GEOMETRIC TECHNIQUES IN GRAPH SEARCH-BASED ROBOT PLANNING Subhrajit Bhattacharya 2012
- 2 TOPOLOGÍA SIN DOLOR SIDNEY A. MORRIS Versión del August 19,2010
- 3 J.A. Bondy and U.S.R. Murty. Graph theory. Graduate texts in mathematics. Springer, 2007.
- 4 Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F.Durrant-Whyte. Information based adaptive robotic exploration. In in Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 540-545, 2002.
- 5 J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. IEEE Trans. Robot. Autom., 20(2):243-255, April 2004.
- 6 Douglas Demyen and Michael Buro. Efficient triangulation-based pathfinding. In AAAI06: Proceedings of the 21st national conference on Artificial intelligence, pages 942 -947. AAAI Press, 2006.
- 7 Edsger W. Dijkstra. A note on two problems in connexion with graphs. Numerische Mathe- matik, 1:269-271, 1959.
- 8 D. Grigoriev and A. Slissenko. Polytime algorithm for the shortest path in a homotopy class amidst semi-algebraic obstacles in the plane. In ISSAC '98: Proceedings of the 1998 international symposium on Symbolic and algebraic computation, pages 17-24, New York, NY, USA, 1998. ACM.
- 9 P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems, Science, and Cybernetics, SSC-4(2):100-107, 1968.
- 10 Allen Hatcher. Algebraic Topology. Cambridge University Press, 2001.
- 11 John Hershberger and Jack Snoeyink. Computing minimum length paths of a given homotopy class. Comput. Geom. Theory Appl, 4:331-342, 1991.
- 12 J. Jost. Compact Riemann Surfaces. Springer-Verlag, 1997.
- 13 Nejat Karabakal and James C. Bean. A multiplier adjustment method for multiple shortest path problem. Technical report, The University of Michigan, June 1995.
- 14 A.G. Kovalev. Vector bundles. 2007. From <http://www.dpmms.cam.ac.uk/agk22/>.
- 15 L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira. Sensing and coverage

- for a network of heterogeneous robots. In Proc. of the IEEE Conf. on Decision and Control, pages 3947-3952, Cancun, Mexico, December 2008.
- 16 Joseph J. Rotman. An Introduction to Algebraic Topology. Springer, 1988.
 - 17 E. Schmitzberger, J.L. Bouchet, M. Dufaut, D. Wolf, and R. Husson. Capture of homotopyclasses with probabilistic road map. In International Conference on Intelligent Robots and Systems, volume 3, pages 2317-2322, 2002.
 - 18 M. Schwager, J. McLurkin, and D. Rus. Distributed coverage control with sensory feedback for networked robots. In Proc. of Robot.: Sci. and Syst., Philadelphia, PA, August 2006.
 - 19 M. Schwager, J. E. Slotine, and D. Rus. Decentralized, adaptive control for coverage with networked robots. In Proc. of the IEEE Intl. Conf. on Robot. and Autom., pages 3289:3294, Rome, Italy, April 2007.
 - 20 C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In Proc. of Robot.: Sci. and Syst., pages 65:72, Cambridge, MA, June 2005.
 - 21 Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, 2005.
 - 22 Benjamn Tovar, Fred Cohen, and Steven M. LaValle. Sensor beams, obstacles, and possible paths. In Workshop on the Algorithmic Foundations of Robotics, pages 317-332, 2008.
 - 23 Ali Ahmadzadeh, James Keller, George J. Pappas, Ali Jadbabaie, and Vijay Kumar. Critical cooperative surveillance and coverage with unmanned aerial vehicles. In Vijay Kumar Oussamma Khatib and Daniela Rus, editors, International Symposium on Experimental Robotics, STAR, Rio de Janeiro, July 2006. Springer-Verlag.
 - 24 Dimitri P. Bertsekas and John N. Tsitsiklis. Parallel and Distributed Computation: Numerical Methods. Prentice Hall, 1989.