

MAPEO Y LOCALIZACION SIMULTANEOS UTILIZANDO UN SISTEMA DE VISION

ANEXOS



**LUIS CARLOS ALVEAR VEGA
LUIS FELIPE VELASCO MUÑOZ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE INVESTIGACIÓN EN APLICACIÓN DE TECNOLOGIAS
INTELIGENTES
POPAYÁN
2008**

**MAPEO Y LOCALIZACION SIMULTANEOS UTILIZANDO UN
SISTEMA DE VISION**

ANEXOS

**Luís Carlos Alvear Vega
Luís Felipe Velasco Muñoz**

**Trabajo de Grado para Optar al Título de
Ingeniero en Automática Industrial.**

**Director
Elena Muñoz España
Ingeniera en Electrónica y Telecomunicaciones**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE INVESTIGACIÓN EN APLICACIÓN DE TECNOLOGIAS
INTELIGENTES
POPAYÁN
2008**

Tabla de contenido

ANEXO A	1
GLOSARIO	1
A.1. Probabilidad Condicional	1
A.2. Teorema de Bayes.....	2
A.3. Distribución normal	2
A.4. Distribución de Probabilidad	3
A.5. Función de Densidad de Probabilidad (FDP).....	4
A.6. Valor Esperado	4
A.7. Varianza.....	5
A.8. Desviación estándar (σ)	6
A.9. Covarianza.....	7
A.10. Matriz de Covarianza	7
ANEXO B	9
PROCESAMIENTO DIGITAL DE IMÁGENES	9
B1.1 Adquisición de La Imagen	10
B.1.1.1. Espacios de color	12
B1.2 Pre-Procesamiento.....	16
B.1.2.1. EL HISTOGRAMA DE UNA IMAGEN.....	17
B.1.2.2. TRANSFORMACION DE INTENSIDAD	17
B.1.2.2.a. BRILLO	18
B.1.2.2.b. CONTRASTE.....	18
B.1.2.2.c. REALCE.....	19
B.1.2.2.d. DETECCION DE BORDES.....	19
B.1.2.2.e. ELIMINACION DE RUIDO	20
B.1.2.3. TRANSFORMACIONES GEOMETRICAS.....	22
B1.3 Segmentación	22
La Segmentación se basa en tres propiedades	23
B.1.3.1. SEGMENTACION BASADA EN BORDES	23

B.1.3.2. SEGMENTACION BASADA EN REGIONES	24
B.1.3.3. SEGMENTACION BASADA EN TECNICAS DE UMBRALIZACION	24
B1.4. Representación y descripción	25
B1.5. RECONOCIMIENTO E INTERPRETACION	26
ANEXO C	26
Modelado UML.....	26
C.1. DIAGRAMA DE CASOS DE USO.....	27
C.1.2. DIAGRAMA DE SECUENCIAS.....	30
C.1.3. DIAGRAMA DE CLASES.....	33
C.1.3.a. CLASE Form1.	34
C.1.3.b. CLASE RECONOCIMIENTO.....	42
C.1.3.c. CLASE SLAM.	46
Bibliografía	57

Lista de Figuras

Figura A 1. Concepto de probabilidad condicional.....	1
Figura A 2. Distribución Gausiana.....	3
Figura B 1. Pasos del procesamiento digital de imágenes [3].	9
Figura B 2. Adquisición digital de imágenes.....	11
Figura B 3. Cámara CCD.	11
Figura B 4. Espacio de color RGB [4].	13
Figura B 5. Representación de una imagen en escala de grises [5].	15
Figura B 6. Representación de una imagen digital [6].....	16
Figura B 7. a. imagen original, b. histograma de la imagen [7].	17
Figura B 8. Mejoramiento del brillo [9].....	18
Figura B 9. Mejoramiento de contraste [9].	19
Figura B 10. Detección de bordes [9].....	20
Figura B 11. Ruido gaussiano [9].....	21
Figura B 12. Ruido impulsional (sal y pimienta) [9].	22
Figura B 13. Transformación geométrica en un plano [9].	22
Figura B 14. Histograma bimodal [11].....	25
Figura C 1. Diagrama de casos de uso del sistema SLAM.	27
Figura C 2. Diagrama de Secuencias del Sistema SLAM.	31
Figura C 3. Diagrama de clases del Sistema SLAM.	34
Figura C 4. Clases relacionadas con la Clase Reconocimiento.....	43

ANEXO A

GLOSARIO

A.1. Probabilidad Condicional

La probabilidad condicional permite determinar la probabilidad de un efecto dada la causa que lo produce.

Generalmente hablando, la probabilidad condicional de un evento A (efecto) dado otro evento B (causa), denotada $P(A/B)$ es la probabilidad de que el evento A ocurra cuando el evento B ha ocurrido. Esta es la razón por la cual se le llama *condicional* a esta probabilidad [1].

Para entender el concepto referirse a la figura A.1.

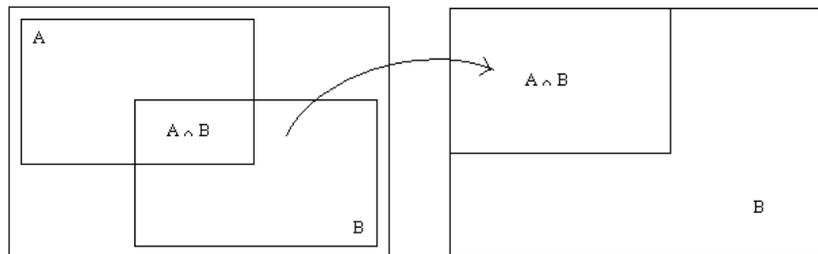


Figura A 1. Concepto de probabilidad condicional.

Si B es cierto, todo lo que caiga fuera de B es imposible, luego B crece hasta ocupar todo el espacio disponible, entonces: $P(B) = 1$ o lo que es lo mismo, $P(B) \rightarrow 1$.

Pregunta: ¿Cuál es la probabilidad de que salga A?

Para responder esta pregunta, se puede observar que solo los casos compatibles con B son importantes, es decir $A \cap B$. de donde se concluye que:

$$P(A \cap B) \rightarrow P(A/B).$$

Luego, haciendo una regla de tres:

$$\begin{array}{ccc} P(B) & \rightarrow & 1 \\ P(A \cap B) & \rightarrow & P(A/B) \end{array}$$

De donde la probabilidad del evento A dado que ha ocurrido el evento B se define como:

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

De otra parte, la probabilidad condicional permite determinar la probabilidad de que ocurra un evento dado que se ha producido otro.

A.2. Teorema de Bayes.

En muchas ocasiones, en lugar de preguntar sobre las probabilidades de los efectos dadas las causas, se hace la pregunta al revés: “sabiendo el efecto, se desea determinar la probabilidad de que la causa haya sido una determinada”. En otras palabras, el teorema de Bayes permite darle la vuelta a la probabilidad condicional [2]. Se define como:

$$P(B/A) = \frac{P(A \cap B)}{P(A)}$$

A.3. Distribución normal

La distribución normal, también llamada distribución de Gauss o distribución gaussiana, es la distribución de probabilidad que con más frecuencia aparece en estadística y teoría de probabilidades [1].

Tiene como característica principal que su función de densidad es simétrica y con forma de campana (figura A.2).

La función de densidad está dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad -\infty < x < \infty$$

Donde μ es la media y σ es la desviación estándar (σ^2 es la varianza) con $\sigma > 0$. Este modelo de probabilidad, conocido como distribución normal o de Gauss, $N(\mu, \sigma)$, de parámetros μ y σ desempeña un papel fundamental en estadística con el propósito de conocer cómo se distribuyen las probabilidades de un evento en relación al resultado del evento.

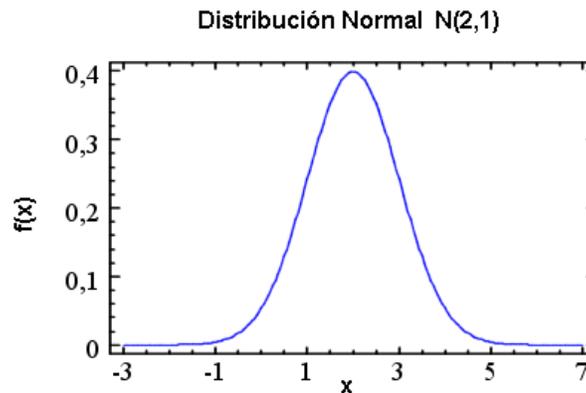


Figura A 2. Distribución Gaussiana.

A.4. Distribución de Probabilidad

Es una función que indica la probabilidad de que una variable aleatoria tome un valor determinado cualquiera, o que pertenezca a un conjunto de valores. La probabilidad en todo el conjunto de valores de la variable aleatoria es igual a 1.

A.5. Función de Densidad de Probabilidad (FDP)

Una función de densidad de probabilidad (FDP) es una función matemática que caracteriza el comportamiento probable de una población.

Es una función $f(x)$ que especifica la posibilidad relativa de que una variable aleatoria continua X tome un valor cercano a x , y se define como la probabilidad de que X tome un valor entre x y $x+dx$, dividido por dx cuando dx es un número infinitesimalmente pequeño. La mayoría de las funciones de densidad de probabilidad requiere uno o más parámetros para especificarlas totalmente [2].

La probabilidad de que una variable aleatoria continua X este ubicada entre los valores a y b está dada por el intervalo de la FDP, $f(x)$, comprendido en el rango entre a y b .

$$\Pr(a \leq x \leq b) = \int_a^b f(x)dx$$

La Función de Distribución de Probabilidad es la derivada (cuando existe) de la función de distribución:

$$f(x) = \frac{dF(x)}{dx}$$

A.6. Valor Esperado

En términos generales, el valor esperado o esperanza, es una medida del valor central en torno al cual tienden a ubicarse los valores muestrales de una distribución de probabilidades.

Sea X una variable aleatoria discreta que toma los valores $X(\Omega) = \{x_1, x_2, \dots, x_n, \dots\}$ con probabilidades p_i , Entonces se define la esperanza matemática de X o valor esperado de X como:

$$\mu = E(x) = \sum xi * p_i$$

$E(x)$ = valor esperado de x .

xi = valores que toma la variable.

p_i = probabilidad asociada a cada uno de los valores de la variable x .

De otra parte, el valor esperado para una variable aleatoria continua X que tenga una función de densidad de probabilidad $f(x)$ es:

$$\mu = E(X) = \int xf(x)dx$$

Esta integral se extiende sobre el intervalo de variación de X .

A.7. Varianza

La varianza, o varianza de la población, es un parámetro de la FDP, que expresa la variabilidad de la población.

La varianza de una variable se define como una medida de dispersión, que es la suma de los cuadrados de las desviaciones de las observaciones de su promedio, dividida por el número de observaciones menos uno.

$$s^2 = \frac{1}{n-1} \sum_i^n (x_i - \bar{x})^2$$

Esta relacionada con la desviación estándar (σ) la cual es la raíz cuadrada de la varianza.

$$\sigma = \sqrt{V(X)} \text{ O bien } \sigma^2 = V(X).$$

A.8. Desviación estándar (σ)

La desviación estándar es una medida del grado de dispersión de los datos del valor promedio. Dicho de otra manera, la desviación estándar es simplemente el promedio o variación esperada con respecto de la media aritmética.

Una desviación estándar grande indica que los puntos están lejos de la media, y una desviación pequeña indica que los datos están cerca de la media [1].

Calculo del promedio o media aritmética \bar{x} .

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Calculo de la desviación estándar (σ)

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Ejemplo del cálculo de la desviación estándar de un conjunto de datos.

Datos {4, 1, 11, 13, 2, 7} edades de un grupo de niños.

Calculamos la media aritmética \bar{x} ,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \text{ con } N=6, x_1=4, x_2=1, x_3=11, x_4=13, x_5=2, x_6=7$$

De donde, $\bar{x} = 6.33$.

Calculamos la desviación estándar σ ,

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \text{ con } N=6, x_1=4, x_2=1, x_3=11, x_4=13, x_5=2, x_6=7$$

$$\sigma = 4.88$$

A.9. Covarianza

La covarianza entre dos variables es una medida de la dependencia recíproca entre dos variables.

La covarianza muestral de una muestra pareada de variables aleatorias X e Y se calcula de la siguiente manera:

$$\sigma_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Donde $x_i, y_i, i = 1, \dots, n$ son elementos de la muestra y \bar{x} y \bar{y} son sus medias muestrales.

σ_{xy} es positiva si los valores altos de x están asociados a los valores altos de y y viceversa.

σ_{xy} es negativa si los valores altos de x están asociados a los valores bajos de y y viceversa.

Si x y y son variables aleatorias independientes entonces $\text{cov}(x, y) = 0$.

A.10. Matriz de Covarianza

Para un conjunto de datos n -dimensional, se puede calcular $\frac{n!}{(n-2)! \cdot 2}$ valores de covarianzas diferentes, puesto que es posible calcular la covarianza entre dos dimensiones cualesquiera.

Una manera útil de tener todas las covarianzas posibles entre las distintas dimensiones, es calcularlas todas juntas y disponerlas en una matriz. La definición para la matriz de covarianza de un conjunto de datos n -dimensional es:

$$C^{n \times n} = (c_i, c_j = \text{cov}(Dim_i, Dim_j))$$

Donde $C^{n \times n}$ es una matriz con n filas y n columnas y Dim_i es la i -ésima dimensión.

ANEXO B

PROCESAMIENTO DIGITAL DE IMÁGENES

En este anexo, se presenta el marco teórico necesario para estructurar los conocimientos básicos que se requieren para entender el procesamiento digital de imágenes.

El análisis humano de imágenes u objetos presenta dificultades, pues el proceso de inspección visual consume una gran cantidad de tiempo y está sujeto a interpretaciones incoherentes e incompletas. El procesado de imágenes, que comprende la manipulación por computador de imágenes reales, resuelve estos problemas al poder automatizar el proceso de extracción de información útil de los datos de la imagen [3]. Con el procesado de imágenes, también se puede perfeccionar la imagen, corregir distorsiones, etc.

A continuación se describen los pasos fundamentales para realizar el procesamiento digital de imágenes. La figura B.1 muestra un esquema de la secuencia de estos pasos.

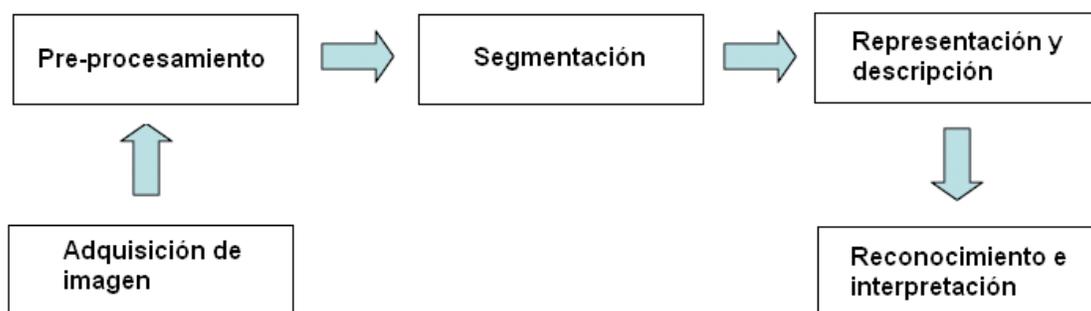


Figura B 1. Pasos del procesamiento digital de imágenes [3].

El primer paso es la adquisición de la imagen; para ello, se requiere de una cámara y la capacidad de digitalizar la señal entregada por ella.

Después de que se ha obtenido la imagen digital, el siguiente paso es su pre-procesamiento. El objetivo de este paso es el mejoramiento de la imagen para su posterior utilización en las etapas siguientes. Algunos métodos utilizados en esta etapa son mejorar el contraste, eliminar el ruido etc.

La siguiente etapa es la segmentación en donde se realiza una partición de la imagen en sus partes constituyentes u objetos. Los datos obtenidos en la etapa de segmentación deben ser convertidos a una estructura apropiada para su posterior procesamiento.

La descripción, también llamada selección de características, extrae información cuantitativa o características relevantes para diferenciar un objeto de los demás.

La última etapa es el reconocimiento e interpretación. El reconocimiento es el proceso que asigna un identificador a un objeto basado en la información provista por sus descriptores. La interpretación es asignar un significado al objeto reconocido. En general, los sistemas de procesamiento que incluyen reconocimiento e interpretación están asociados a aplicaciones de análisis de imágenes cuyo objetivo es la extracción automática de información [3].

A continuación se realiza una descripción más detallada de cada una de las etapas anteriormente señaladas.

B1.1 Adquisición de La Imagen

Tres elementos son necesarios para adquirir imágenes digitales. El primero es un dispositivo físico que sea sensible a la banda visible del espectro electromagnético y que produzca una señal eléctrica proporcional al nivel de energía percibido. El segundo elemento es el medio por el cual la señal eléctrica es transmitida, y el

tercero es un digitalizador, encargado de convertir la señal eléctrica del dispositivo físico a una forma digital como se puede apreciar en la figura B.2.

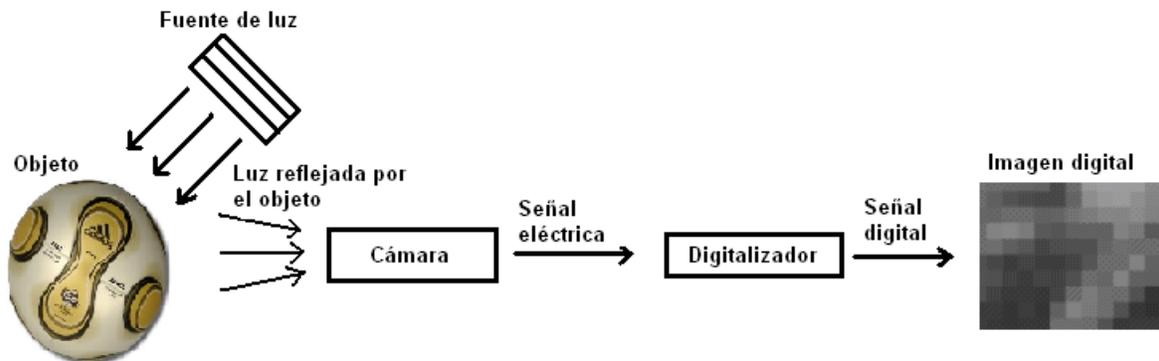


Figura B 2. Adquisición digital de imágenes.

Las cámaras o sensores de imágenes (figura B.3) son los elementos encargados de captar la información luminosa de la escena, y transmitirla al computador como una señal analógica o digital.



Figura B 3. Cámara CCD.

Para la transmisión de la señal eléctrica entregada por la cámara es necesaria la utilización de un cable. Existen muchos tipos, pero los más populares son el S-Video y el video compuesto [3].

Una vez transmitida la señal eléctrica entregada por la cámara, es necesaria una tarjeta de adquisición y procesamiento de imágenes la cual se encarga de convertirla a una señal digital y de procesarla.

Las tarjetas digitalizadoras de imágenes muestrean la señal de la cámara y guardan la imagen digital en una zona de memoria, que puede ser accedida por el microprocesador del computador para su tratamiento.

De otra parte, una imagen capturada es un conjunto de valores numéricos. La resolución es la medida básica de cuanta información es visible en una imagen. Generalmente es descrita en términos de “h”x”v”. Donde “h” es la resolución horizontal y “v” es la resolución vertical. Entre mayor sea la resolución la imagen tendrá un mayor detalle.

Algunas resoluciones posibles son: 160x120, 320x240, 640x480, 1280x960. De otra parte, cada imagen esta compuesta de elementos individuales conocidos como píxeles. Por ejemplo, si se tiene una resolución de 640x480 en realidad se dice que se tienen 480 líneas horizontales y 640 píxeles por cada línea, entonces, la imagen tendría un total de 307.200 píxeles.

Existen distintos formatos para la representación de los valores del color de forma digital y estos son conocidos de manera general como espacios de color.

B.1.1.1. Espacios de color

Un espacio de color es un modelo para la representación matemática de un conjunto de colores. Se les llama espacios de color porque son representaciones en 2, 3 o 4 dimensiones. Cada punto dentro del espacio corresponde a un color. Los tres espacios de color más utilizados son el RGB, YUV y HSI [3].

B.1.1.1.1. RGB

Este espacio de color tiene tres componentes: Rojo, Verde y Azul que corresponden a los tres colores primarios (los componentes individuales son mezclados para formar el color deseado) y son representados en un sistema cartesiano tridimensional como se muestra en la figura B.4 [4].

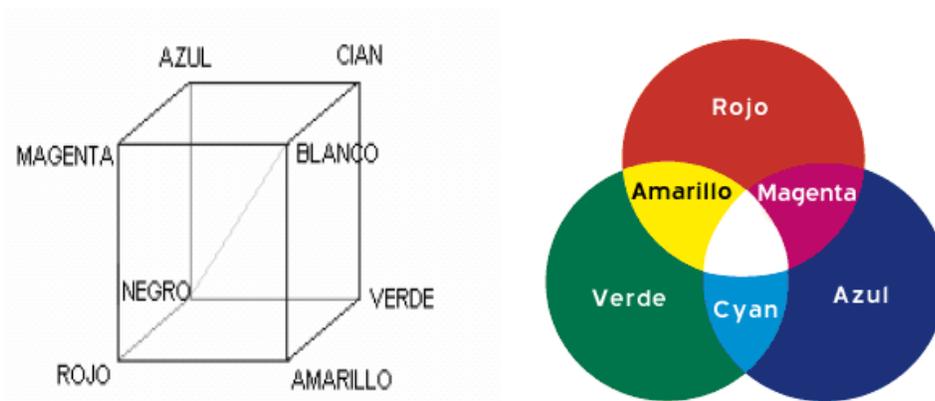


Figura B 4. Espacio de color RGB [4].

Este espacio de color es generalmente utilizado en procesamiento de imágenes, pero presenta una desventaja para aplicaciones de visión por computadora debido a que no se puede separar la luminosidad del color, es decir, se necesita de tres componentes para poder obtener la luminosidad¹, haciendo difícil la adaptación de la aplicación a condiciones de luz variable.

B.1.1.1.2. YUV.

Para hacer compatible el cambio de televisión en blanco y negro a televisión a color se creó el espacio de color YUV [4].

¹ La luminosidad hace referencia a la cantidad de luz recibida sobre una superficie.

La luminosidad (Y) es la parte de la señal de video relacionada a la cantidad de brillo en cualquier punto de la imagen. Si la luminosidad es alta, la imagen es brillante, y si es baja, la imagen es oscura, es decir, el nivel de gris en la imagen.

La cromaticidad (U y V) se define como la diferencia entre un color y una referencia blanca con la misma cantidad de luminosidad. U se obtiene al restar la luminosidad al componente azul ($U = B - Y$) y V al componente rojo ($V = R - Y$) donde B es el componente azul y R el componente rojo.

El espacio de color YUV al tener codificada la cromaticidad en dos componentes (U y V) y la luminosidad en un tercero (Y) puede ser útil para aplicaciones de visión porque pueden adaptarse a condiciones de luz variable al ignorar el componente de luminosidad sin tener impacto en la detección de colores [4].

B.1.1.1.3. HSV.

El espacio de color HSV fue creado para poder manejar el color de manera intuitiva y fue diseñado en base a la manera en que los humanos percibimos e interpretamos el color.

El matiz (H) es lo que normalmente se piensa como color. La saturación (S) es la cantidad de blanco que es mezclado con el color. Una saturación de cero indica que no hay matiz y el color esta dentro de la escala de grises. La componente V indica la luminosidad.

El espacio de color HSV presenta las mismas ventajas que el espacio de color YUV para aplicaciones de visión debido a la fácil separación entre luminosidad y cromaticidad, sin embargo, al no estar soportado directamente por hardware el manejo de este espacio de color puede resultar en un procesamiento muy costoso en tiempo[4].

B.1.1.2. REPRESENTACION DIGITAL DE LA IMAGEN EN ESCALA DE GRISES.

En computación, una escala de grises es una escala empleada en las imágenes digitales en las que el valor de cada píxel posee un valor equivalente a una graduación de gris. Las imágenes representadas de este tipo están compuestas de sombras de grises, que van desde el negro más profundo variando gradualmente en intensidad de grises hasta llegar al blanco como se puede ver en la figura B.5 [5]



Figura B 5. Representación de una imagen en escala de grises [5].

Con el término imagen monocroma² o simplemente imagen, se hace referencia a una función de dos dimensiones $f(x,y)$, donde x e y indican las coordenadas espaciales, y el valor de la función en el punto (x,y) es proporcional al brillo (nivel de gris o intensidad) de la imagen en ese punto. Una imagen digital es una función $f(x,y)$ que ha sido discretizada en ambas coordenadas espaciales (x,y) y en el nivel de gris (figura B.6). Las imágenes en escala de grises, emplean 8 bits para representar cada píxel lo que permite una escala con 256 intensidades o niveles de gris.

² Imagen monocromática: es aquella que contiene píxeles en blanco o negro no grises o de color.

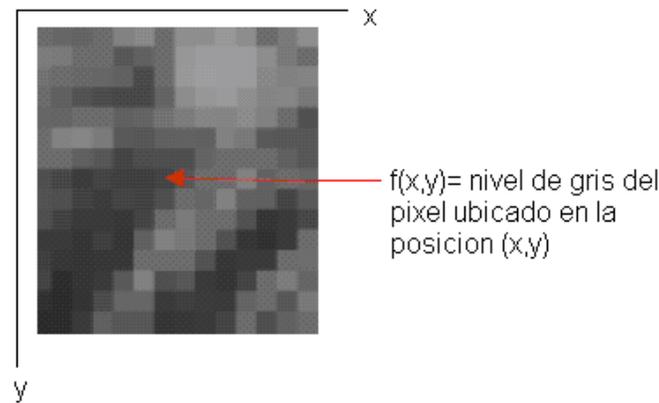


Figura B 6. Representación de una imagen digital [6].

B1.2 Pre-Procesamiento

El pre-procesamiento es el segundo paso del procesamiento digital de imágenes. Se le da este nombre a las operaciones en las imágenes al más bajo nivel de abstracción, es decir, la entrada y salida de este paso son imágenes con valores de intensidad similares a los datos originales capturados por el digitalizador [4].

El pre-procesamiento de imágenes no incrementa la cantidad de información de una imagen, sin embargo, es muy útil porque ayuda a suprimir información que no es relevante para los objetivos particulares de análisis en un caso dado. Por lo tanto, el objetivo del pre-procesamiento es una mejora de los datos de la imagen que suprima las distorsiones indeseadas e incremente las características relevantes para su posterior procesamiento.

Para simplificar el desarrollo de los distintos algoritmos para el procesado, éstos normalmente se realizan sobre imágenes previamente pasadas a grises, concretamente a 256 niveles de grises, por lo que el rango de valores de la función $f(x,y)$ es:

$$0 \leq f(x, y) \leq 255$$

Desde un punto matemático, se pretende desarrollar el tipo de transformaciones que se deben aplicar a una imagen para obtener otra con mejores características. Es decir:

$$g(x, y) = T(f(x, y))$$

B.1.2.1. El histograma de una imagen.

El histograma de una imagen contiene la información de la probabilidad de aparición de las distintas tonalidades de color que se pueden presentar, ya que se puede trabajar en distintos tipos de colores o en escala de grises como lo indica la figura B.7 [7].

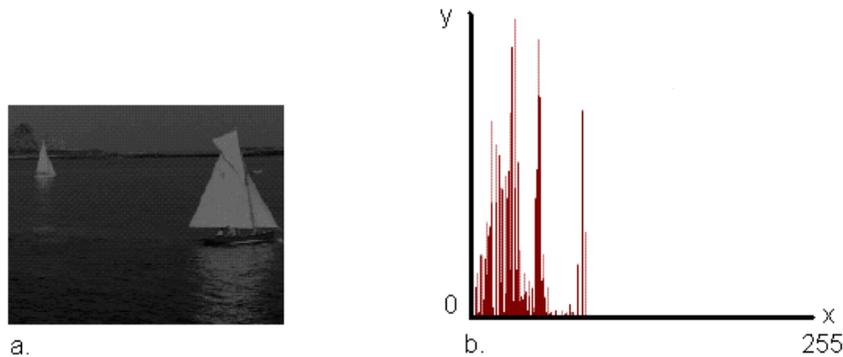


Figura B 7. a. imagen original, b. histograma de la imagen [7].

Las intensidades o niveles de gris están representadas a lo largo del *eje x* y el número de ocurrencias para cada intensidad se representan en el *eje y*. Algunos métodos de pre-procesamiento son las transformaciones de intensidad y las transformaciones geométricas.

B.1.2.2. Transformación de intensidad.

En una transformación de intensidad se modifica el valor del píxel sin importar su posición dentro de la imagen, algunas transformaciones de intensidad incluyen modificar el brillo, contraste, matiz, saturación, gamma, etc.

B.1.2.2.a. BRILLO

El brillo hace referencia a la cantidad de luz que emite la imagen. De otra parte el brillo en imágenes en escala de grises indica el nivel de gris o intensidad que poseen los píxeles de una imagen, es decir, es el valor de $f(x,y)$ en cada punto de la imagen. Valores grandes de brillo representan tonos claros, mientras que valores pequeños representan tonos oscuros como lo muestra la figura B.8.

Variar el nivel de brillo de una imagen consiste en sumar o restar a todos los píxeles un nivel constante de gris. La expresión que calcula el brillo de una imagen de niveles de grises es la siguiente [4]:

$$g(x, y) = f(x, y) + \text{brillo}$$

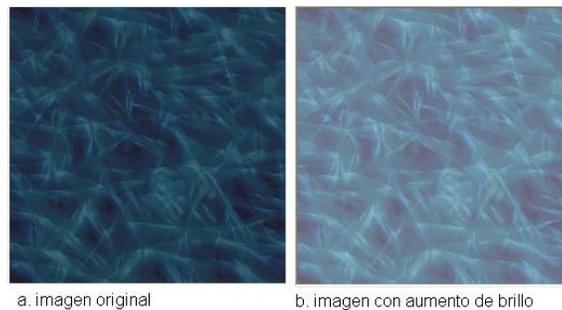


Figura B 8. Mejoramiento del brillo [9].

B.1.2.2.b. CONTRASTE

El contraste hace referencia a que tan lejos están los valores más blancos de los valores más negros en una imagen. Si el valor más blanco está muy lejos del valor más negro se dice que la imagen tiene un alto contraste [4]. Con lo cual la imagen es muy nítida como lo indica la figura B.9. Si los dos valores están muy cercanos el uno del otro se dice que la imagen tiene un pobre o bajo contraste y no se puede distinguir la diferencia entre blanco y negro provocando que la imagen aparezca grisácea.

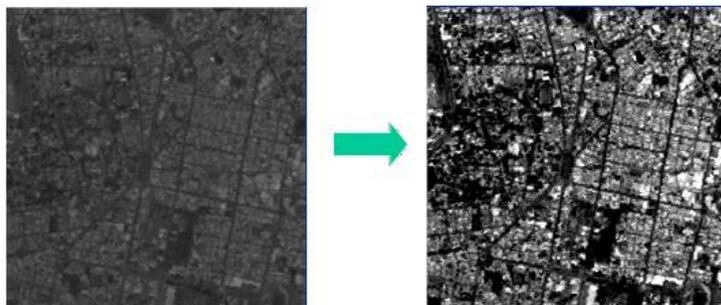


Figura B 9. Mejoramiento de contraste [9].

B.1.2.2.c. REALCE

El objetivo principal del realce es el de destacar los detalles finos de una imagen o intensificar detalles que han sido difuminados, bien sea por error o bien por efecto natural del método de adquisición de la imagen. El realce de una imagen aumentará su contraste.

B.1.2.2.d. DETECCION DE BORDES

Uno de los más importantes y sencillos procesos es la detección de bordes, porque de él se puede empezar a extraer información de la imagen, como pueden ser las formas de los objetos que las componen.

La detección de bordes es a menudo el primer paso en la segmentación de imágenes, y se utiliza para agrupar los píxeles en regiones para determinar una composición de la imagen [6].

Los bordes de una imagen contienen mucha de la información de la imagen. Los bordes cuentan donde están los objetos, su forma, su tamaño, y también sobre su textura. Los ejes o bordes se encuentran en zonas de una imagen donde el nivel de intensidad fluctúa bruscamente, cuanto más rápido se produce el cambio de intensidad, el eje o borde es más fuerte. Un buen proceso de detección de bordes

facilita la elaboración de las fronteras de objetos con lo que, el proceso de reconocimiento de objetos se simplifica.

En general, los bordes de objetos en una imagen se pueden distinguir por los cambios más o menos bruscos de valor entre dos o más píxeles adyacentes como lo indica la figura B.10.

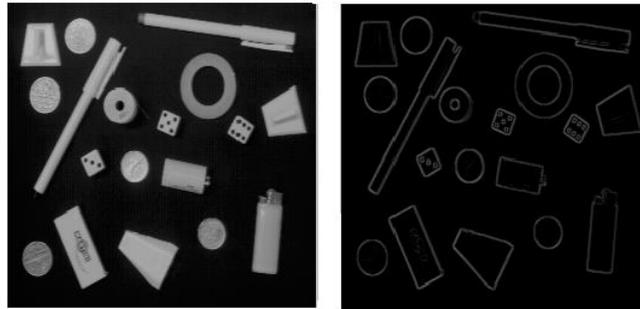


Figura B 10. Detección de bordes [9].

La diferencia entre los valores de los píxeles indica lo acentuado del borde, de forma que a mayores diferencias se tienen bordes más marcados y a menores bordes suavizados.

B.1.2.2.e. ELIMINACION DE RUIDO

Se entiende por ruido en imágenes digitales cualquier valor de un píxel de una imagen que no se corresponde exactamente con la realidad. Cuando se adquiere una imagen digital normalmente está contaminada por ruido [4]. El ruido se debe, la mayoría de las veces al equipo electrónico utilizado en la captación de las imágenes (ruido de cuantificación de la imagen, efecto de niebla en la imagen... etc).

De otra parte, todas las imágenes tienen una cierta cantidad de ruido, valores distorsionados, bien debidos al sensor CCD de la cámara o al medio de

transmisión de la señal. El ruido se manifestará generalmente en píxeles aislados que toman un valor de gris diferente al de sus vecinos.

El ruido puede clasificarse de dos tipos [4]:

- **Gaussiano:** Produce pequeñas variaciones en la imagen. Es debido, por ejemplo, a las diferentes ganancias en el sensor, ruido en los digitalizadores, perturbaciones en la transmisión, etc. Este tipo de ruido se puede observar en dos imágenes tomadas sucesivamente como lo indica la figura B.11. En apariencia son iguales, pero si se obtiene la resta puede comprobarse como existe una diferencia entre las dos. Se considera siempre, por tanto, que el valor final del píxel sería el ideal más una cantidad correspondiente al error, este error puede describirse como una variable gaussiana.

$$I[i, j] = I_{ideal}[i, j] + p$$

Donde

$$p \approx N(0, \sigma^2)$$

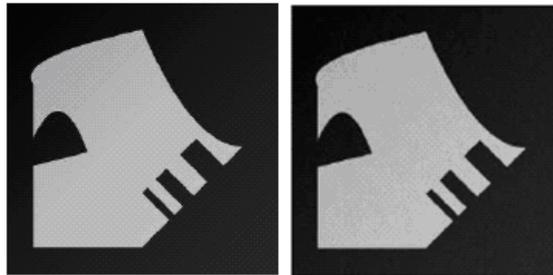


Figura B 11. Ruido gaussiano [9].

- **Impulsional (conocido como sal y pimienta):** El valor que toma el píxel no tiene relación con el valor ideal sino con el valor de ruido que toma valores muy altos o muy bajos. Se caracteriza entonces porque el píxel toma un valor máximo, causado por una saturación del sensor, o mínimo, si se ha perdido su señal [3].

Dicho de otra manera existe la ocurrencia de píxeles completamente blancos y píxeles completamente negros como lo ilustra la figura B.12.

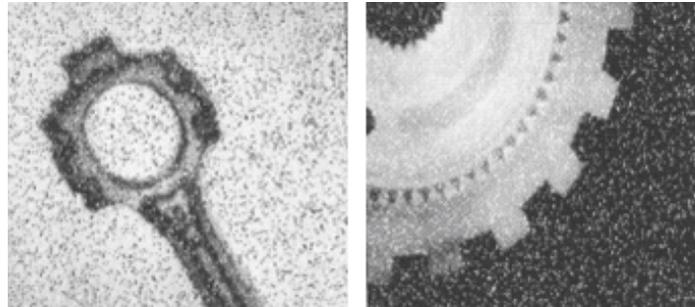


Figura B 12. Ruido impulsional (sal y pimienta) [9].

B.1.2.3. Transformaciones geométricas.

Las transformaciones geométricas son comunes en gráficas por computadora y también son utilizadas en análisis de imágenes. Permiten la eliminación de distorsiones geométricas que ocurren cuando una imagen es capturada, como por ejemplo las provocadas por los lentes de la cámara.

Una transformación geométrica es una función T que coloca el píxel (x, y) en una nueva posición (x', y') . La figura B.13 ilustra el concepto.

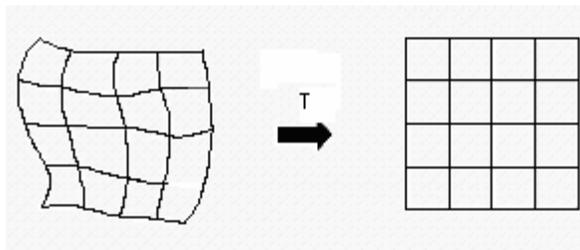


Figura B 13. Transformación geométrica en un plano [9].

B1.3 Segmentación

El tercer paso del procesamiento digital de imágenes es la segmentación que subdivide a una imagen en sus partes u objetos constituyentes. El nivel al que se lleva esta subdivisión depende del problema que se este resolviendo, esto es, la segmentación debe detenerse cuando los objetos de interés para una aplicación han sido aislados [10].

En general, la segmentación automática es una de las tareas más difíciles en el procesamiento de imágenes. Este paso determina el eventual éxito o fracaso del análisis de la imagen.

La Segmentación se basa en tres propiedades

- Similitud: cada uno de los píxeles de un elemento tienen valores parecidos para alguna propiedad.
- Discontinuidad: los objetos destacan del entorno y tienen por tanto unos bordes definidos.
- Conectividad: los píxeles pertenecientes al mismo objeto tienen que ser contiguos, es decir, deben estar agrupados. La conectividad de los píxeles desempeña un papel importante en la segmentación de las imágenes.

Los algoritmos de segmentación de imágenes tienen tres formas comunes: métodos basados en bordes, técnicas basadas en regiones y técnicas de umbral.

B.1.3.1. Segmentación basada en bordes.

Los métodos basados en bordes se centran en la detección de contornos. Delimitan el borde de un objeto y segmentan los píxeles dentro del contorno como pertenecientes a ese objeto. Su falencia consiste en conectar contornos separados o incompletos, lo que los hace susceptibles a fallas.

B.1.3.2. Segmentación basada en regiones.

Las técnicas basadas en regiones, usualmente operan de la siguiente forma: la imagen es dividida en regiones agrupando píxeles vecinos con niveles de intensidad similares. Las regiones adyacentes son unidas bajo cierto criterio que involucra la homogeneidad de las fronteras de la región [10].

B.1.3.3. Segmentación basada en técnicas de umbralización.

Las técnicas de umbral segmentan la imagen píxel a píxel, es decir, no toman en consideración el valor de los píxeles vecinos para el proceso. Si el valor de un píxel esta dentro del rango de valores especificado para un objeto el píxel es segmentado. Estas técnicas son efectivas cuando los objetos y el fondo de la imagen tienen rangos de valores diferentes y existe un contraste marcado entre ellos.

Con la umbralización se convierte una imagen con varios niveles de gris a una nueva con solo dos, de manera que los objetos queden separados del fondo. Para realizar la binarización, basta con explorar todos los puntos de la imagen y determinar que puntos corresponden al objeto y cuales al fondo. La técnica más simple para realizar esto es particionar el histograma de la imagen usando un umbral simple T . La imagen particionada $g(x, y)$ esta definida como [11]:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) \geq T \\ 0 & \text{si } f(x, y) < T \end{cases}$$

Entonces los píxeles etiquetados con 1 corresponden a los objetos, mientras que los píxeles etiquetados con 0 corresponden al fondo [11].

La selección del valor del umbral, se realiza generalmente a partir del histograma de la imagen. Así si una imagen esta compuesta de un objeto que aparece en la escena sobre un fondo, es de esperar que el histograma sea bimodal, es decir, si por ejemplo el objeto es más claro que el fondo, pues en el histograma aparecerán dos picos, el ubicado en los valores de gris más elevados correspondiente al objeto y otro pico para niveles de gris más bajos, correspondientes al fondo [7].

En la figura B.14 se muestra un histograma bimodal, en el cual el umbral se ubica entre los dos picos del histograma.

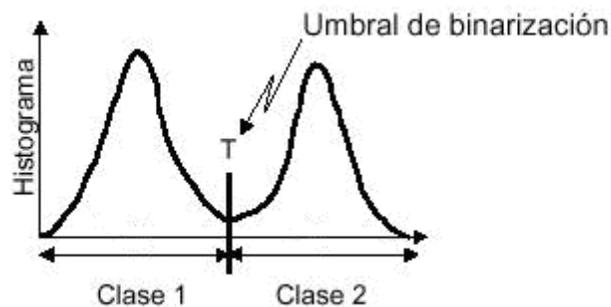


Figura B 14. Histograma bimodal [11].

B1.4. Representación y descripción

Después que una imagen ha sido segmentada en regiones, el conjunto de píxeles segmentados son usualmente representados y descritos en una estructura adecuada para su posterior procesamiento. La representación es conveniente porque en lugar de procesar una gran cantidad de datos se procesa una estructura sencilla que contiene la misma información [4].

Se puede representar la región en términos de sus características externas (frontera) o bien, representarla en términos de sus características internas (los píxeles que componen a la región). Una vez escogida la representación de la región esta debe ser descrita de acuerdo a la decisión tomada. Por ejemplo, una región puede ser representada por su frontera y descrita por características como su longitud, la orientación de la línea recta entre sus puntos extremos, etc.

Generalmente, una representación externa es escogida cuando el enfoque principal esta en las características de la figura. Una representación interna es seleccionada cuando interesan las propiedades de los píxeles.

B1.5. RECONOCIMIENTO E INTERPRETACION

El último paso del procesamiento digital de imágenes es el reconocimiento e interpretación que consiste en descubrir, identificar y entender patrones que son relevantes en el desempeño de una tarea dada.

Un patrón es una descripción cuantitativa o estructural de un objeto o entidad en una imagen. En general, un patrón esta formado por uno o más descriptores (también conocidos como características) y son agrupados en clases. Una clase es una familia de patrones que comparten propiedades comunes [4]. El reconocimiento de patrones comprende técnicas para asignar cada patrón a su respectiva clase de manera automática y sin intervención humana.

ANEXO C

Modelado UML

En el proceso de modelado UML (*Unified Modeling Language*) del Sistema de Localización y Mapeo Simultáneos se utilizaron tres tipos de diagramas, el diagrama de casos de usos quien describe el sistema de forma general, el diagrama de secuencias que muestra la dinámica del sistema y el diagrama de clases, donde se muestra el tipo de interacción estática entre las clases que componen el sistema.

C.1. DIAGRAMA DE CASOS DE USO.

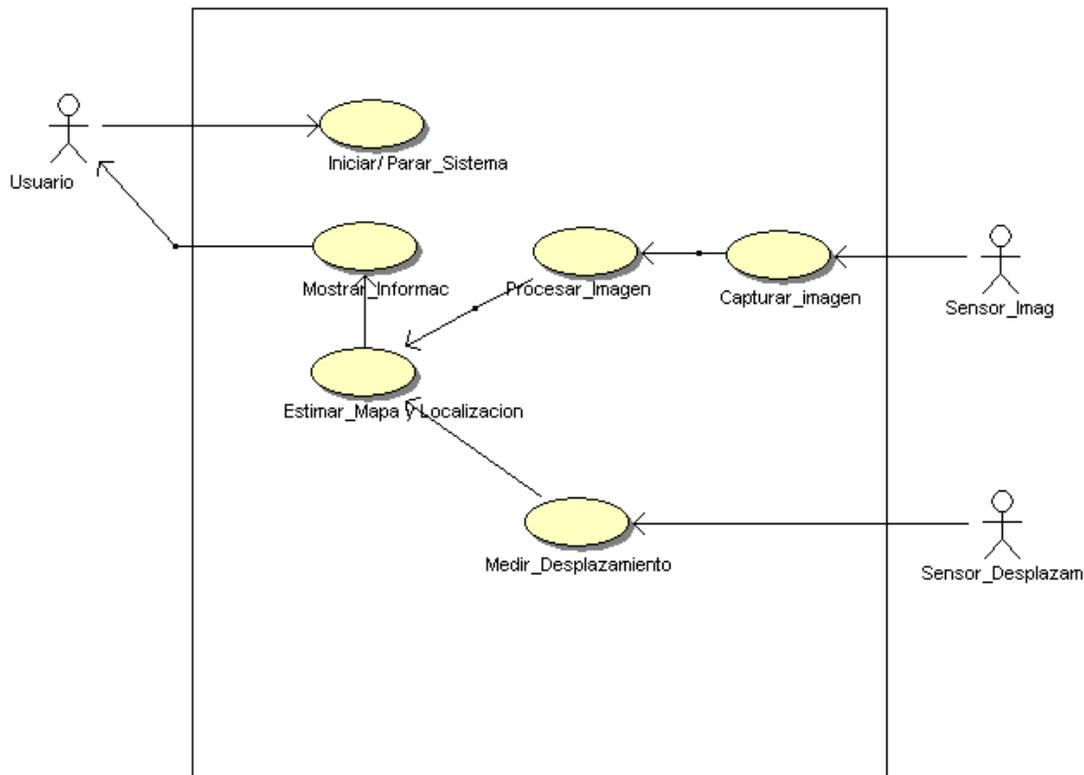


Figura C 1. Diagrama de casos de uso del sistema SLAM.

En el modelo de casos de uso se definió como frontera la aplicación desarrollada en C#, se identificaron 3 actores quienes interactúan directamente con la aplicación y se definieron 6 casos de uso con los cuales se cumple el proceso de Localización y Mapeo Simultáneos utilizando un sistema de Visión (figura C1). A continuación se presenta una descripción de cada uno de estos componentes.

Frontera:

Aplicación en C# que realiza el proceso de SLAM utilizando una *cámara web* como sensor de visión y *encoders incrementales* como dispositivos de medición de desplazamiento del robot.

Actores:

- *Usuario*: Es el ente que da inicio y parada al sistema, además es quien observa todo el proceso de adquisición de datos y Localización y Mapeo Simultáneos. Por ultimo es quien configura los parámetros de la cámara utilizada y ajusta los parámetros de funcionamiento del sistema.
- *Sensor de Imágenes*: Es quien se encarga de capturar las imágenes del entorno. Estas imágenes son la base para extraer las características del entorno, es decir, extraer las marcas presentes en el entorno. Como sensor de imágenes en este trabajo se utilizó una cámara web con conexión USB.
- *Sensor de Desplazamiento*: Este actor hace referencia a los sensores propioceptivos ubicados en el móvil (*encoders*) los cuales entregan información sobre el desplazamiento del robot dentro del ambiente recorrido.

Casos de uso:

- *Iniciar/parar sistema*: El usuario es quien da inicio al sistema, configura los parámetros de la cámara y del sistema en general y es quien decide en que momento dar por terminado el proceso de Localización y Mapeo Simultáneos.
- *Medir Desplazamiento*: Este caso de uso, cada cierto tiempo se muestrea la posición del robot leyendo los datos generados por los *encoders* incrementales dispuestos en las ruedas del robot.

- *Capturar imagen:* Una vez muestreada la posición del robot, se procede a la captura de la imagen del entorno, a partir de la cual se extraen las marcas presentes en el ambiente a través del cual se desplaza el robot.
- *Procesar imagen:* Dentro de este caso de uso se incluye el procesamiento de las imágenes capturadas, con el fin de extraer de ellas la información más importante de cada objeto detectado en cada imagen vista por el robot.
- *Estimación del mapa y localización:* Se encarga de realizar el proceso de estimación del mapa y localización del robot dentro de este, utilizando la información que proviene de los sensores de desplazamiento y del sensor de características del entorno, para éste caso una cámara web.
- *Mostrar información:* Representa la interfaz de trabajo la cual es la encargada de gestionar todos los componentes de la aplicación para realizar el proceso de SLAM, además esta interfaz es la encargada de mostrar la evolución del mapa a medida que el robot se desplaza por el entorno y debe mostrar toda la información extraída de éste.

Con los anteriores componentes del diagrama de casos de uso se definió el comportamiento del sistema como sigue:

Cuando el usuario decide dar inicio al sistema, tanto el sensor de captura de imágenes como los sensores de desplazamiento se ponen en funcionamiento; una vez el móvil comienza su desplazamiento a través del ambiente estructurado, se recibe la información de los sensores cada cierto tiempo, la información entregada por la cámara de visión es procesada para extraer de ella la información relevante que permita ir construyendo el mapa del ambiente que recorre el móvil, a la vez la información entregada por los *encoders* ubicados en las ruedas del móvil permitirá inferir la posición actual del móvil.

Con la información obtenida, el sistema debe ser capaz de reconstruir el ambiente recorrido por el móvil, además de determinar la localización actual del móvil dentro del ambiente.

De otra parte, el usuario debe tener la opción de observar a cada instante como va evolucionando la construcción del mapa así como la localización actual del móvil dentro del ambiente en estudio.

Es importante mencionar que el usuario puede parar el sistema en cualquier instante, sin afectar con ello la evolución que haya tenido el sistema hasta ese momento.

C.1.2. DIAGRAMA DE SECUENCIAS.

En este diagrama se muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que se intercambian ordenados según su secuencia en el tiempo. La figura C.2 muestra el diagrama de secuencias planteado para la aplicación de Localización y Mapeo Simultáneos utilizando un sistema de visión.

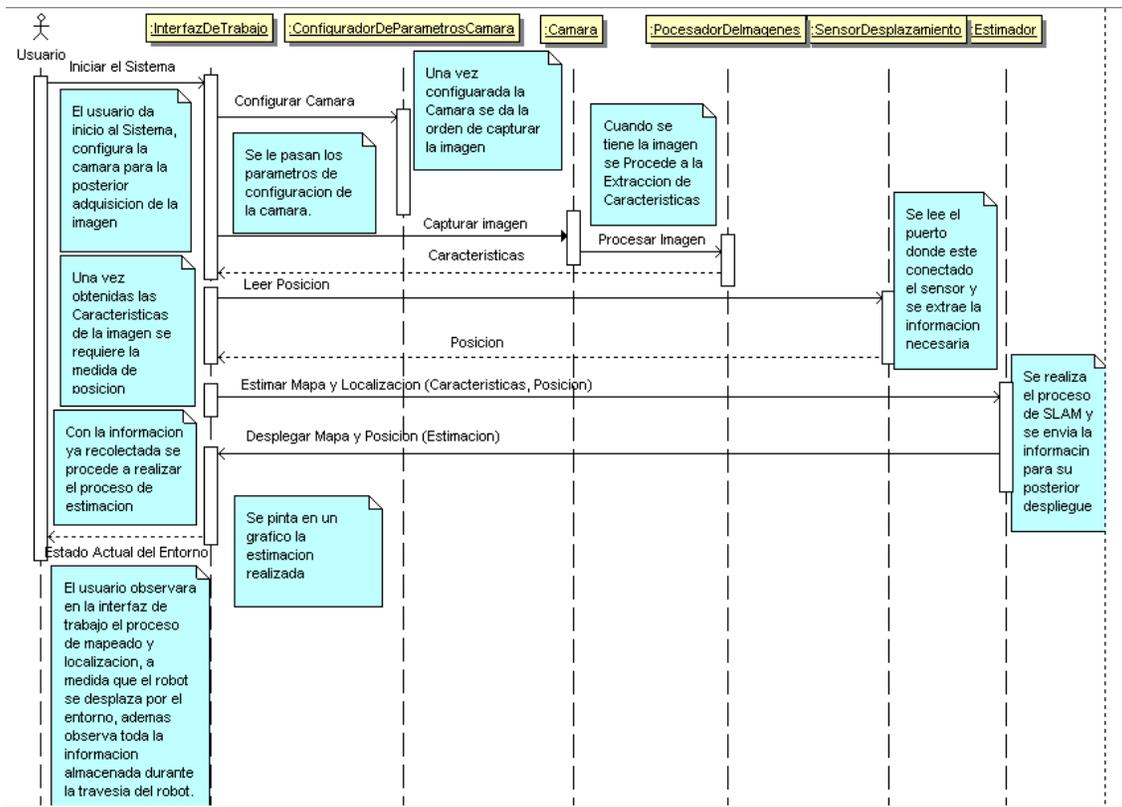


Figura C 2. Diagrama de Secuencias del Sistema SLAM.

En este diagrama de secuencias se plantea un actor que interactúa con seis objetos.

El actor usuario es el ente encargado de darle inicio o parada al sistema, además tiene la posibilidad de observar una interfaz de trabajo donde se presentara la evolución del proceso de mapeado y localización, a medida que el robot se desplaza por el entorno, los seis objetos con los cuales interactúa se describen a continuación.

- *Interfaz de trabajo:* Este objeto es el encargado de interactuar con el usuario, además gestiona la recepción de información desde los sensores de desplazamiento como del dispositivo de medición de características. Otra de sus funciones es gestionar la estimación del mapa y la posición del robot administrando la información mencionada anteriormente, una vez realizada

esta estimación, la interfaz también se encarga de mostrar el mapa estimado junto con la posición del robot y todos los datos generados durante este proceso.

- *Configurador de parámetros de la cámara:* Este objeto permite realizar la configuración de los parámetros de trabajo de la cámara. Antes de comenzar el proceso de Localización y Mapeo Simultáneos es necesario realizar la configuración del tipo de cámara a utilizar, la velocidad de adquisición y el tamaño de la imagen.
- *Cámara:* Cámara es un objeto que realiza las labores de captura de la imagen, este objeto almacena la imagen en un variable de tipo mapa de bits para su posterior procesamiento.
- *Procesador de imágenes:* Este objeto toma la imagen de la variable de tipo mapa de bits y le realiza el procesamiento adecuado para realzar los objetos de interés, segmentar, identificar y caracterizar cada una de las marcas presentes en la imagen.
- *Sensor de desplazamiento:* Este objeto es el encargado de generar información correspondiente al desplazamiento del robot, entrega medidas de desplazamiento en x, y y la orientación que tiene el robot.
- *Estimador:* El objeto estimador es quien aplica el algoritmo EKF utilizando los datos de odometría y características del entorno para realizar el proceso de Localización y Mapeo Simultáneos, almacena en una variable global la información correspondiente a las características extraídas del entorno y a la posición del robot dentro de este.

C.1.3. DIAGRAMA DE CLASES.

El diagrama de clases muestra las diferentes clases que componen el sistema y cómo se relacionan unas con otras. Estos diagramas son “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

El sistema desarrollado está constituido a partir de 3 clases principales:

1. Form1.
2. Reconocimiento.
3. SLAM.

La figura C.3 muestra el diagrama de clases del Sistema de Localización y Mapeo Simultáneos mediante un sistema de visión desarrollado. Cabe aclarar que las variables utilizadas en nuestro desarrollo se definieron en su mayoría como tipo *doubles*, con el fin de estandarizar el tipo de variables y para evitar la continua conversión de tipos al realizar operaciones entre variables, debido a que *double* es el tipo de variables que permite trabajar en un amplio rango y con el manejo de decimales que es indispensable al obtener los resultados de las operaciones en las diferentes etapas del EKF.

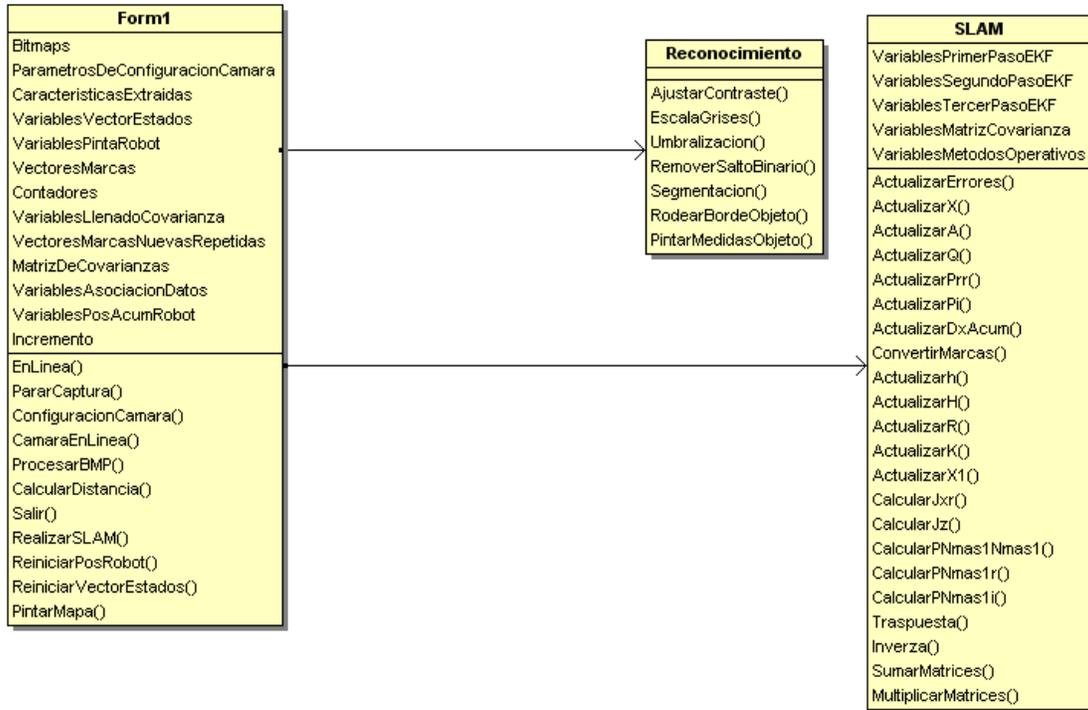


Figura C 3. Diagrama de clases del Sistema SLAM.

C.1.3.a. CLASE Form1.

Esta clase es la encargada de gestionar los métodos de las clases Reconocimiento y SLAM para lograr el proceso de Localización y Mapeo Simultáneos utilizando una cámara web como sistema de visión. Form1 es la clase donde se realiza la programación de la interfaz grafica, tanto la parte visual como la parte lógica, es además desde donde se gestionan los métodos de configuración de la cámara, los métodos de captura, mejoramiento y procesamiento de las imágenes. Una vez realizados los procesos anteriormente mencionados, la clase Form1 se encarga de gestionar las operaciones correspondientes al proceso de SLAM mediante los métodos de la clase respectiva, utilizando la información anteriormente extraída. Por ultimo esta clase se encarga de mostrar toda la información recogida durante el proceso de SLAM mediante graficas que muestran la evolución del sistema y mediante regiones donde se muestran los datos calculados.

Los atributos con los que cuenta la clase Form1 son los siguientes:

- *Bitmaps.*

Este atributo es un fichero de tipo *Bitmap*, donde se almacenan cada una de las imágenes en sus diferentes etapas de procesamiento.

- *Parámetros de configuración de la cámara.*

Esta sección esta formada por varios atributos de diferentes tipos, estos son:

ImgPorSeg: Variable de tipo *double* donde se almacena el valor correspondiente al número de imágenes por segundo mediante el cual se realizara la captura de las imágenes con la cámara web.

UmbAreaMin, UmbAreaMax: En esta se almacena el valor de área mínimo y máximo de área que puede tener un objeto en la imagen para que sea detectado como marca del entorno, esta es de tipo *double*.

AnguloVision: Variable de tipo *double* donde se guarda el valor del ángulo de visión de la cámara, en el programa este valor puede ser modificado dependiendo de las características de la cámara con que se trabaje.

Xlinea1, Xlinea2: Estas son dos variables de tipo *double* donde se almacenan posiciones de puntos que sirven como guías para pintar el ángulo de visión de la cámara con que se trabaje, estos valores también son modificables.

- *Características Extraídas.*

Esta sección esta conformada por diferentes atributos que se utilizan para la extracción de características, los cuales se muestran a continuación.

VectorAlto, VectorAncho: Vector de tipo *double* donde se almacenan las medidas de altura y de ancho de cada marca extraídas de la imagen.

VectorDistanciaMarcas: Vector de tipo *double* donde se almacenan las medidas de distancia directa cámara-marca.

VectorPx: Vector de tipo *double* donde se almacenan las posiciones de las marcas en el eje horizontal de la imagen.

VectorAnguloTotal: Vector de tipo *double* donde se almacenan los ángulos a los cuales se observan las marcas con respecto al eje x de la cámara.

VectorAnguloMedio: Vector de tipo *double* donde se almacenan los ángulos a los cuales se observan las marcas con respecto al eje y de la cámara, es decir, al centro de la imagen capturada.

VectorPosX, VectorPosY: Vectores de tipo *double* donde se almacenan las posiciones en X, Y de las marcas con respecto al plano del robot.

VectorArea: Vector de tipo *double* se almacenan las medidas de área de todas las marcas observadas en la imagen. Cabe anotar que el área es calculada como la suma de píxeles que componen la imagen.

Cont: Variable de tipo *double* que lleva las posiciones de los vectores de características extraídas de las marcas, es decir, representan el número de marcas a las cuales se le extrajeron características.

VectorDist: Vector de tipo *double* donde se almacenan las distancias entre ejes paralelos marcas-cámara, con las cuales se calcula *VectorDistanciaMarcas*.

- *Variabes Vector de Estados.*

Estas variables son usadas para realizar la actualización del vector de estados general de sistema.

VectorEstadosX: Vector de tipo double donde se almacenan tanto la posición y orientación del robot como las posiciones de las marcas observadas.

VectorEstadosX1, VectorEstadosX2, VectorEstadosX11, VectorEstadosX21:
Vectores de tipo *double* que se utilizan como auxiliares para realizar procesos de llenado y actualización del vector de estados general del sistema.

- *Variables para Pintar el Robot.*

Variables con las cuales se generan las coordenadas x, y de los tres puntos que sirven como guía para pintar el triángulo que representa el robot en el plano global.

Xtriangulo, Ytriangulo, Xtriangulo1, Ytriangulo1, Xtriangulo2, Ytriangulo2:
Variables de tipo *double* que permiten establecer la ubicación x, y de los puntos que forman el triángulo que representa al robot en el panel2.

XtrianguloA, YtrianguloA, Xtriangulo1A, Ytriangulo1A, Xtriangulo2A, Ytriangulo2A: Variables de tipo *double* que permiten establecer la ubicación x, y de los puntos que forman el triángulo que representa al robot en el panel3.

- *Vectores Marcas.*

Variables donde se almacenan las posiciones x, y de las marcas en el plano global.

Xmarca, Ymarca, XmarcaB, YmarcaB: Vectores de tipo *double* donde se almacenan las posiciones x, y de todas las marcas vistas, a partir de estos vectores se realiza la asociación de datos y el llenado del vector de estados global del sistema.

- *Contadores.*

Variables donde se llevan los conteos de los diferentes eventos que ocurren en el sistema.

Cont1: Variable de tipo int donde se almacena el número de marcas vistas por ciclo de muestreo, es decir, las marcas presentes en cada imagen.

Cont2: Variable de tipo int donde se almacena el número de marcas acumulado durante todos los muestreos de las imágenes.

n1: Variable de tipo int donde se lleva el conteo de marcas nuevas vistas con respecto al instante inmediatamente anterior.

n2: Variable de tipo int donde se lleva el conteo de las marcas repetidas vistas con respecto al instante inmediatamente anterior.

- *Variables Llenado Matriz Covarianza.*

Variables auxiliares que permiten realizar el llenado de la matriz de covarianzas del sistema.

K1, N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12, N13, N14, N15: Variables de tipo int mediante las cuales se puede hacer el llenado de cada uno de los diferentes componentes que conforman la matriz de covarianzas en su respectiva posición dentro de esta matriz.

- *Vectores Marcas Nuevas y Repetidas.*

Vectores donde se almacenan las marcas catalogas como nuevas o repetidas después del proceso de asociación de datos.

Vector Marcas Repetidas X, Vector Marcas Repetidas Y: Vector de tipo *double* donde se almacenan las posiciones X, Y de las marcas que fueron vistas en el instante inmediatamente anterior y que son vistas en el estado actual.

Vector Marcas Nuevas X, Vector Marcas Nuevas Y: Vector de tipo *double* donde se almacenan las posiciones X, Y de las marcas que no fueron vistas en el instante inmediatamente anterior y que son vistas en el estado actual.

- *Matriz de Covarianzas.*

Matriz de tipo *double* donde se almacenan las covarianzas de la posición del robot, las covarianzas de las marcas nuevas observadas, las correlaciones entre las posiciones de las marcas y la posición del robot y viceversa y las correlaciones cruzadas entre marcas.

- *Variables Asociación de Datos.*

Estas variables son utilizadas para hacer posible el desarrollo del proceso de asociación de datos a medida que se observan nuevas marcas.

Xtemp, Ytemp: Estos son dos vectores de tipo *double* que almacenan temporalmente las posiciones x, y de cada marca observada, con el fin de compararlas con las marcas observadas en el instante siguiente y así determinar cuales marcas han sido vistas en el instante inmediatamente anterior y cuales son recientemente observadas.

Resta: Resta es un vector de tipo *double* donde se almacena temporalmente la diferencia entre las posiciones de las marcas anteriores y las marcas actuales, esta variable se define con el propósito de observar el comportamiento de comparación de posiciones de las marcas.

- *Variables Posición Acumulada del Robot.*

Estas variables se definen con el fin de ayudar a llevar el registro de los desplazamientos que tuvo el robot a través del entorno.

VectorDxAcum: Vector de tipo double donde se almacena en cada una de sus posiciones los deltas acumulados de movimientos en x, y realizados por el robot.

VectorXacum, VectorYacum: En cada una de las posiciones de estos vectores de tipo double se almacenan los últimos deltas de movimiento x, y realizados por el robot.

- *Incremento.*

Variable de tipo *int* encargada de llevar el conteo del número de veces que se han muestreado los sensores del robot, tanto los *encoders* incrementales como las características extraídas mediante la cámara.

Los métodos con los que cuenta la clase Form1 son los siguientes:

- *Configuración Cámara.*

Este método se encarga de invocar el panel de configuración de la cámara propio de la aplicación instalada para realizar las capturas de las imágenes en C# llamada IC Imaging Control.

- *Cámara en Línea.*

Este método da inicio al proceso de captura de las imágenes desde la cámara web, almacena cada imagen temporalmente en la posición cero de un fichero de *Bitmaps* para realizar el posterior procesamiento, hasta que en un instante siguiente esta imagen se reemplace por la nueva imagen capturada.

- *Parar Captura.*

Este método detiene el proceso de captura de imágenes, abortando el proceso de almacenamiento de las imágenes en la memoria disponible del PC para la captura de imágenes.

- *Procesar BMP.*

Método que se encarga de gestionar los métodos de la clase Reconocimiento para realizar el proceso de captura y mejoramiento de la imagen, además del proceso de extracción de características de la misma.

- *Calcular Distancia.*

Este método utiliza las características extraídas en el anterior método para realizar las conversiones respectivas de estas características a términos de distancia con el cual finalmente se realizará el proceso de SLAM.

- *Realizar SLAM.*

Método que se encarga de gestionar los métodos de la clase SLAM para realizar el proceso de SLAM basado en el filtro extendido de Kalman.

- *Reiniciar pos Robot, Reiniciar Vector Estados.*

Métodos que se encargan de regresar a sus valores iniciales, las variables correspondientes a la posición del robot y al vector de estados respectivamente.

- *Pintar Mapa.*

Método encargado de realizar las graficas que representan el mapa global del entorno sobre el cual se desplaza el robot y la posición actual que éste ocupa dentro del espacio.

- *Salir.*

Método que da salida inmediata del sistema.

C.1.3.b. CLASE Reconocimiento.

Esta clase se encarga de tomar la imagen capturada en un *Bitmap* y realizarle los procesos de mejoramiento y pre-procesamiento, para luego realizar las labores de extracción de características de todas las marcas observadas en cada imagen. Esta clase se relaciona con las clases indicas en la figura C.4 para realizar los procesos anteriormente mencionados.

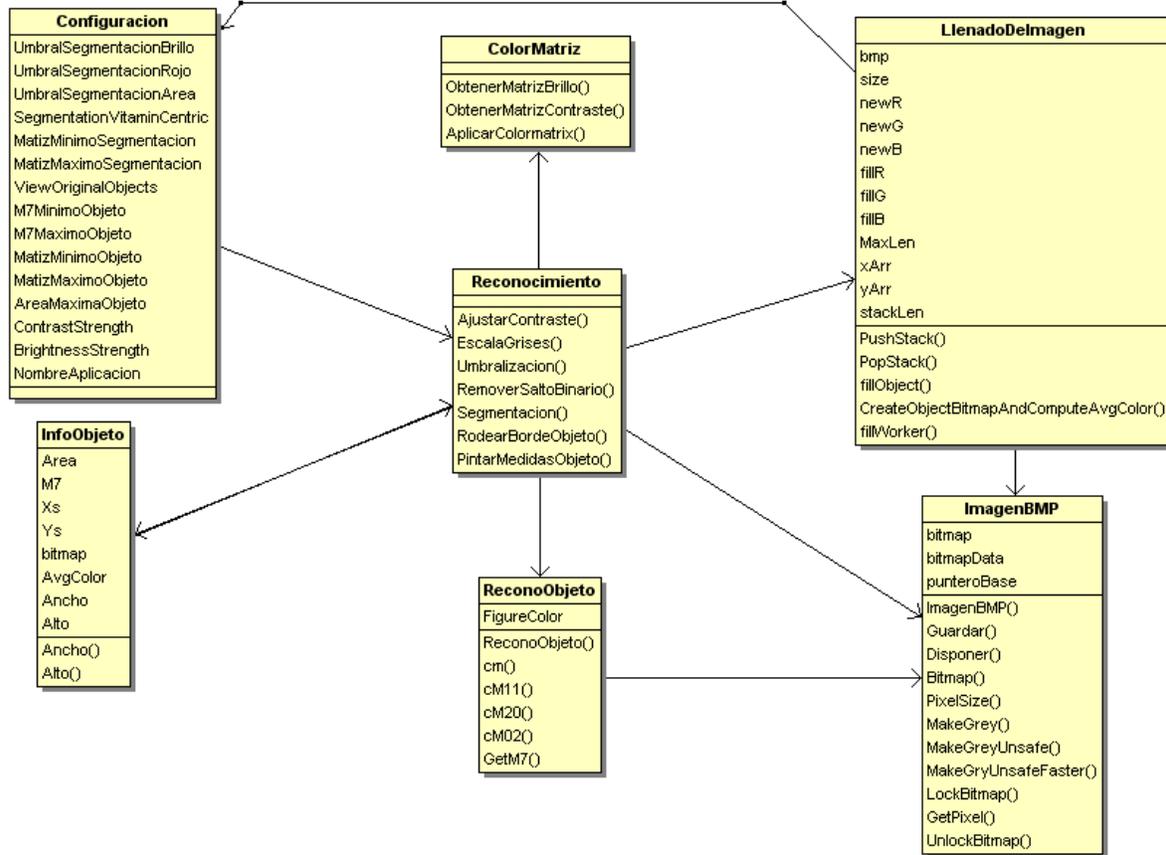


Figura C 4. Clases relacionadas con la Clase Reconocimiento.

En la clase Configuración se definen todos los parámetros que se utilizarán para las diferentes fases del procesamiento de imágenes como son mejoramiento de la imagen, segmentación e identificación.

La clase ColorMatriz contiene los métodos con los cuales se elabora el proceso de mejoramiento de contraste de la imagen.

La clase ImagenBMP es una clase del fichero System.Drawing, contiene una estructura llamada PixelData, esta agrupa los atributos R, G, B e incorpora un método *Brightness* con estos atributos. Luego se define el *Bitmap* en el programa,

se determina su tamaño, con esta información se aplican los métodos *MakeGrey*, *MakeGreyUnsafe* y *MakeGreyUnsafeFaster*.

Dentro de la clase *LlenadoDeImagen*, se realiza el llenado de un *Bitmap*, correspondiente al mapa binario de la imagen a partir de los objetos encontrados.

En la clase *ReconoObjeto* se realiza el proceso de reconocimiento de objetos, se les asigna una etiqueta y se realizan algunas operaciones sobre éstos para determinar sus características.

La clase *InfoObjeto* contiene toda la información referente a los objetos: Área, Xs, Ys, ancho, largo.

Una vez mencionadas las clases con las cuales tiene relación la clase *Reconocimiento*, a continuación se describirán sus métodos dado que esta no contiene atributos, pues utiliza su función es operar los atributos de las clases con que se relaciona.

- *Ajustar Contraste.*

Este método toma el *Bitmap* presente en la posición cero del fichero de *Bitmaps* y le realiza el proceso de mejoramiento de contraste con el fin de realzar las diferencias entre los objetos y el fondo de la imagen. Este método retorna un *Bitmap* que contiene la imagen original, pero con el contraste mejorado.

- *Escala de Grises.*

Luego de ajustar el contraste, se procede a la conversión del *Bitmap* a escala de grises para facilitar la diferenciación entre los objetos y el entorno. A este método se le entrega la imagen generada luego del ajuste de contraste, y este retorna una imagen en escala de grises, la cual posteriormente será binarizada para la detección de objetos.

- *Umbralización.*

Una vez llevada la imagen a escala de grises, se procede a la binarización de la misma. Dentro de este método se establece un umbral, se recorre la imagen píxel por píxel y el que este por encima de ese umbral fija su valor al máximo (255), y el que este por debajo de ese umbral se fija su valor al mínimo (0), de esta manera se diferencia totalmente el objeto del fondo.

- *Remover Salto binario.*

Luego de umbralizar la imagen, se aplica el filtro de “sal y pimienta” para eliminar los posibles ruidos en los bordes de cada objeto.

- *Segmentación.*

Por ultimo se recorre la imagen y se realiza la segmentación de cada uno de los objetos presentes en ella. A este método se le pasa como argumento la imagen filtrada y este retorna una imagen donde se observan cada uno de los objetos diferenciados mediante un cuadro rojo que se muestra alrededor de cada objeto. En este punto se almacena cada una de las características de

cada objeto, ancho, alto y área, posición en x y posición en y, en la clase InfoObeto, con las cuales en pasos siguientes se inferirán las medidas de distancia y ángulo de cada marca.

C.1.3.c. CLASE SLAM.

Por ultimo, la clase SLAM es la encargada de realizar la estimación de la posición del robot y de las marcas observadas por este, realizando una serie de operaciones matriciales que permiten actualizar las variables del proceso de SLAM mediante el Filtro Extendido de Kalman EKF. Esta clase esta compuesta por los atributos necesarios para llevar el registro del estado del sistema como son el vector de estados, la matriz de covarianzas y las jacobianas de los modelos de movimiento y de medida. Además, esta clase cuenta con los métodos que hacen posible llevar a cabo los tres pasos principales del EKF como son la actualización del estado actual mediante datos de odometría, la actualización del estado actual mediante las marcas re-observadas y la incorporación de nuevas marcas al estado actual.

Los atributos con los que cuenta la clase SLAM son los siguientes:

Variables Primer Paso EKF.

- $X, Xtemp.$

Estos son vectores de tipo *double*, los cuales son utilizados para almacenar en ellos tanto la posición y orientación del robot como la posición de las marcas observadas en el entorno. $Xtemp$ se utiliza para almacenar el estado anterior del sistema, mediante la comparación de este vector con el vector X se puede definir

las marcas nuevas vistas en el estado actual y las marcas re-observadas en el estado anterior.

- Dx .

Vector de tipo *double* donde se almacenan los últimos movimientos en distancia y orientación del robot.

- $A, ATaux, AT$.

Matrices de tipo *double* las cuales son usadas para actualizar la matriz jacobiana del modelo de predicción A . AT es la matriz donde se almacena la transpuesta de A y $ATaux$ es una matriz auxiliar que ayuda en el cálculo de la transpuesta de A .

- Q .

Matriz de tipo *double* donde se define el ruido del proceso.

- C .

Constante de tipo *double* donde se define un valor que representa que tan exactas son las medidas tomadas por los sensores de desplazamiento.

- $Prod1, Prod2, Prod3, Prod4, PrrAux, Prr$.

Matrices de tipo *double* mediante las cuales se actualiza la parte superior izquierda de la matriz de covarianzas, es decir, la covarianza de la posición del robot Prr en el primer paso del EKF.

- $Pri, PriAux$.

Matrices de tipo *double* que sirven para actualizar la correlación cruzada entre la posición del robot y la *i*-ésima marca observada P_i .

Variables Segundo Paso EKF.

- *Z.*

Vector de tipo *double* donde se almacenan las observaciones de las marcas tomadas del entorno.

- *Lambda.*

Vector de tipo *double* donde se almacenan las posiciones x , y de las marcas repetidas observadas.

- *Grados.*

Variable auxiliar de tipo *double* donde se almacena la conversión de grados a radianes, debido a que la función que calcula las funciones trigonométricas como seno y coseno tiene como entrada el valor del ángulo en radianes.

- *v.*

Vector de tipo *double* donde se cargan los valores del error en la medida de las marcas del entorno.

- *h.*

Vector de tipo *double* donde se guarda el modelo de medida de cada marca re-observada con respecto al plano general.

- *HTAux, HT, H.*

Matrices de tipo *double* mediante las cuales se actualiza la matriz jacobiana del modelo de predicción H.

- *Raux, R, Raux1, R1.*

Raux y R son matrices de tipo *double* donde se define el error en la medición de las marcas re-observadas. Raux1, R1 son matrices de tipo *double* donde se define el error en la medición de las marcas nuevas.

- *VTAux, VAux, VAux1, V, VT.*

Matriz identidad V la cual es utilizada para operar la matriz R con otras matrices, VAux, VAux1, VTAux son matrices auxiliares en las cuales se almacenan las diferentes operaciones que se van realizando sobre V y R.

- *Kprod, Kprod1, Kprod2, Kprod3, Kprod4, Kprod5, Kprod6, Ksum, Ksum1, K.*

Matrices de tipo *double* las cuales sirven para actualizar la ganancia de Kalman K.

- *Det, INV, INV1.*

Det es una variable tipo *double* la cual es usada para almacenar el determinante de una matriz, INV e INV1 son matrices de tipo *double* donde se almacenan las inversas de matrices que son necesarias de calcular durante el proceso de SLAM EKF.

- X_{prod}, X_{prod1} .

Vectores de tipo *double* mediante los cuales se actualiza el estado actual del robot utilizando las marcas re-observadas en el entorno.

- **Variables Tercer Paso EKF.**

J_x, J_xTras, J_xTras1 : Matrices de tipo *double* mediante las cuales se actualiza la matriz jacobiana de la marca con respecto a la posición del robot x, y , y su transpuesta que representa la jacobiana de la posición del robot con respecto a la posición de la marca.

J_z, J_zTras, J_zTras1 : Matriz de tipo *double* donde se actualiza la matriz jacobiana de la marca con respecto a su ángulo y magnitud en su plano de ubicación y su transpuesta que representa la jacobiana de la ubicación de la marca en términos de ángulo y magnitud con respecto a la posición de la marca.

$PN_{mas1}N_{mas1}$: Matriz de tipo *double* donde se almacena la covarianza de cada nueva marca.

PN_{mas1r}, PN_{mas1r1} : Matrices de tipo *double* mediante las cuales se actualiza la correlación entre la nueva marca y la posición del robot.

$PrrJ_x, PrrJ_xTras, PrrJ_xTras1, J_xPrrJ_xT, J_xPrrJ_xT1, RJ_zTras, RJ_zTras1, J_zRJ_zTras, J_zRJ_zTras1, J_xPrrJ_xTMasJ_zRJ_zTras, J_xPrrJ_xTMasJ_zRJ_zTras1$: Matrices de tipo *double* mediante las cuales se actualiza la correlación entre la nueva marca y la posición del robot, cuando hay más de una marca nueva observada.

$PriTras, PriTras1, J_xPri, J_xPri1, PN_{mas1i}Tras, PN_{mas1i}Tras1$: Matrices de tipo *double* mediante las cuales se actualiza la correlación entre marca y marca, y la

transpuesta de esta correlación que significa la correlación en sentido contrario entre marcas.

JxrAcum, *JxrPriAcum*, *JxrPri1Acum*, *PNmas1iTrasAcum*, *PNmas1iTras1Acum*,
JxrAcum1, *JxrPriAcum1*, *JxrPri1Acum1*, *PNmas1iTrasAcum1*,
PNmas1iTras1Acum1, *JxrAcum2*, *JxrPriAcum2*, *JxrPri1Acum2*,
PNmas1iTrasAcum2, *PNmas1iTras1Acum2*, *JxrAcum3*, *JxrPriAcum3*,
JxrPri1Acum3, *PNmas1iTrasAcum3*, *PNmas1iTras1Acum3*: Matrices de tipo *double* mediante las cuales se actualiza la correlación entre maraca y marca cuando se observan mas de dos marcas, y la transpuesta de esta correlación que significa la correlación en sentido contrario entre marcas.

- **Variables Matriz Covarianzas.**

TetamasLamdaTeta: Variable de tipo *double* donde se almacena la suma del ángulo al cual se observa una marca mas el ángulo al cual se encuentra el robot, con el fin de referenciar esta nueva marca al plano global.

DeltaT: Variable de tipo *double* donde se lleva el registro de la magnitud del desplazamiento del robot.

XacumuladoAux, *YacumuladoAux*, *XacumuladoAux2*, *YacumuladoAux2*,
XacumuladoAux3, *YacumuladoAux3*: Variables utilizadas para llevar el registro de los desplazamientos del robot, su función es contribuir al cálculo de la correlación entre una marca nueva observada y las marcas observadas en estados anteriores del sistema.

DxAcumulado, *DXAcumtemp*: Vectores de tipo *double* donde se almacenan los desplazamientos acumulados en cada paso.

Xacumulado, Yacumulado: Vectores de tipo *double* donde se lleva el registro de los movimientos en X e Y del robot, con el fin de pintar la trayectoria recorrida por el robot a través del entorno.

- ***Variables Métodos Operativos.***

M1, M2, M3, M11, M22, M33, M111, M111a, M222, M222a, M333, M333a, M333b: Matrices de tipo *double* las cuales son utilizadas como matrices de entrada y salida en los diferentes métodos para realizar las diferentes operaciones matriciales como suma, resta, multiplicación, inversa...etc.

Los métodos con los que cuenta la clase SLAM son los siguientes:

A continuación se presentan los métodos involucrados en el desarrollo del primer paso del EKF.

- *Actualizar Errores.*

En este método se cargan las constantes del error en la medida de las marcas del entorno.

- *Actualizar X.*

Este método realiza la actualización el estado actual mediante las variaciones en los sensores de odometría.

- *Actualizar A.*

En este método se actualiza la matriz jacobiana del modelo de predicción A.

- *Actualizar Q.*

Este método realiza la actualización de la matriz de error del proceso Q.

- *Actualizar Prr.*

En este método se actualiza la parte superior izquierda de la matriz de covarianza P, es decir, la covarianza para la posición del robot.

- *Actualizar Pi.*

En este método se realiza la actualización la parte superior de la matriz de covarianza P, es decir, la covarianza de la posición del robot vs marcas nuevas.

- *Actualizar DXAcum.*

Este método realiza la actualización del vector de desplazamientos acumulados. A continuación se presentan los métodos involucrados en el desarrollo del segundo paso del EKF.

- *Convertir Marcas.*

Este método convierte las marcas medidas en magnitud y ángulo a coordenadas cartesianas.

- *Actualizar h.*

En este método se realiza la estimación de la distancia y el ángulo de cada marca re-observada.

- *Actualizar H.*

Este método actualiza la jacobiana del modelo de medida H.

- *Actualizar R.*

En este método se actualiza la matriz de error de medida R.

- *Actualizar K.*

En este método se realiza la actualización de la ganancia de Kalman k, mediante la cual se realizara la corrección de la posición del robot.

- *Actualizar X1.*

En este método se realiza la actualización del estado del sistema X para cada marca re-observada.

A continuación se presentan los métodos involucrados en el desarrollo del tercer paso del EKF.

- *Calcular Jxr.*

Este método realiza el cálculo de la jacobiana de la nueva marca con respecto a la posición actual del robot.

- *Calcular Jz.*

En este método se realiza el calculo de la jacobiana de la marca con respecto a su posición x, y.

- *Calcular PNmas1Nmas1.*

Este método realiza el cálculo de la covarianza para la nueva marca.

- *Calcular PNmas1r.*

En este método se calcula la correlación entre la nueva marca y la posición actual del robot.

- *Calcular PNmas1i.*

En este método se calcula la correlación entre la marca nueva y la anterior, este método es la estructura principal para calcular la correlación entre una marca nueva adherida al sistema y las existentes en el estado global del sistema.

Por ultimo se presentan los métodos involucrados en el desarrollo de las diferentes operaciones matriciales necesarias para desarrollar cada uno de los pasos del EKF.

- *Traspuesta.*

Método que realiza la obtención de la traspuesta de una matriz, dentro de la aplicación se desarrollaron varios métodos que realizan este proceso para matrices de diferentes tamaños.

- *Inversa.*

Método que realiza la obtención de la inversa de una matriz.

- *Sumar Matrices.*

Método que realiza la suma de dos matrices, en la aplicación se desarrollaron varios métodos que realizan este proceso para matrices de diferentes tamaños.

- *Multiplicar Matrices.*

Método que realiza la multiplicación de dos matrices, en la aplicación se desarrollaron varios métodos que realizan este proceso para matrices de diferentes tamaños.

Bibliografía

[1] DEFINICIONES DE PROBABILIDAD. Disponible en:

http://www.uam.es/personal_pdi/psicologia/carmenx/EsquemaTema15.pdf.

[2] DÍAZ, RAFAEL. "Introducción a la Probabilidad, los Procesos Estocásticos y la Estadística en Ingeniería". Octubre, 2003. Disponible en:

<http://www.neutron.ing.ucv.ve/electronica/materias/c2508/cap%202.pdf>

[3] MARTINEZ GOMEZ, LUIS ALFREDO. "Sistema de Visión para El Equipo de Robots Móviles del ITAM". Instituto Tecnológico Autónomo de México. México D.F. 2004.

[4] ESCALERA HUESO, ARTURO. "Visión por Computador, Fundamentos y Métodos". Pearson Educación. S.A. Madrid, 2001.

[5] ESCALA DE GRISES-WIKIPEDIA. La Enciclopedia Libre. Disponible en: http://es.wikipedia.org/wiki/Escala_de_grises. Fecha de actualización: 22 de febrero de 2008. Fecha de acceso: 11 de marzo de 2008.

[6] C. GONZÁLEZ, RAFAEL and WOODS, RICHARD E." Tratamiento digital de imágenes".

[7] CAMPUS VIRTUAL, Enseñanza Virtual y Laboratorios Tecnológicos."Tratamiento Digital de la Imagen, El Histograma de una imagen". Disponible en: http://campusvirtual.uma.es/tdi/www_netscape/TEMAS/Tdi_30/. Fecha de actualización: 8 de marzo de 2008. Fecha de acceso: 4 de marzo de 2008.

[8] GRUPO DE TECNOLOGIA INDUSTRIAL. "Características de una Imagen". Universidad Miguel Hernández. Visión por Computador. División de Ingeniería de Sistemas y Automática.

[9] HERNANDEZ HOYOS, MARCELA. "Procesamiento y análisis de Imágenes Digitales (PAID)". Compugráfica. 2004.

[10] CASTRO PEREZ, DAVID y DOPICO RODRIGUEZ, PATRICIA. "Segmentación, Umbralización, Regiones y Clustering". Universidad Carlos III. Madrid, 2004.

[11] MARTIN, MARCOS. "Técnicas Clásicas de Segmentación de Imagen". Mayo, 2002.

Direcciones de Internet de interés

<http://gps-tsc.upc.es/imatge/Main/TEI/segmentacion.pdf>

<http://delta.cs.cinvestav.mx/~fraga/Cursos/Rec3D/segmentacion.pdf>