

**“APLICACIÓN DE FILTROS CROMÁTICOS EN EL
ESPACIO
DE COLOR PARA LA DETECCIÓN Y MEDICIÓN DE
FORMAS EN PLANTAS DE SOJA”.**

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA Y COMPUTACIÓN

CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO FINAL DE GRADO:

ALUMNO: GUILLERMO H. FARINA

DIRECTORES DEL PROYECTO: ING. GUILLERMO N. ABRAS / DR.
JUAN IGNACIO PASTORE

LABORATORIO DE PROCESAMIENTO DIGITAL DE IMÁGENES.



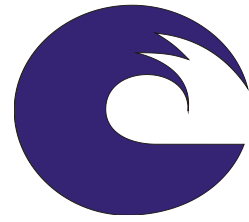
RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



**“APLICACIÓN DE FILTROS CROMÁTICOS EN EL
ESPACIO
DE COLOR PARA LA DETECCIÓN Y MEDICIÓN DE
FORMAS EN PLANTAS DE SOJA”.**

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA Y COMPUTACIÓN

CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO FINAL DE GRADO:

ALUMNO: GUILLERMO H. FARINA

DIRECTORES DEL PROYECTO: ING. GUILLERMO N. ABRAS / DR.
JUAN IGNACIO PASTORE

LABORATORIO DE PROCESAMIENTO DIGITAL DE IMÁGENES.

CAPÍTULO I

INTRODUCCIÓN

Capítulo I

INTRODUCCIÓN

1.1. Motivación.

El cultivo de soja ha ido creciendo en los últimos años hasta transformarse en la actualidad en uno de los puntales de la economía argentina. Debido a la gran rentabilidad obtenida por su exportación, hay un aumento sostenido de la superficie destinada a esta producción en comparación con otros cultivos.

Este aumento en la producción trae como consecuencia la necesidad de profundizar la investigación científica y el desarrollo tecnológico en la rama de las ciencias agrícolas, con el objetivo de optimizar la productividad en las áreas sembradas.

En este contexto, el INTA Balcarce presentó la patente de un robot que ayudará a obtener variedades de soja más tolerantes a la falta de agua, la limitación más frecuente en los rendimientos de este cultivo.

El objetivo de este desarrollo es aumentar la escala y precisión en la medición de las características físicas de las plantas (el fenotipo) lo que permite a los investigadores relacionarlas con la constitución genética de cada individuo (el genotipo), para acelerar la obtención de nuevas variedades mejoradas de los cultivos.

La búsqueda de nuevos genes que pudieran ser utilizados para el mejoramiento de la soja tiene como intención primordial que las nuevas variedades sean más resistentes a las enfermedades, utilicen menos agroquímicos y usen en forma más eficiente el agua. En definitiva, se busca que el cultivo de soja sea más sostenible económica y energéticamente. El robot, que permite someter a las plantas a diferentes situaciones hídricas, puede simular desde condiciones de exceso de agua hasta sequías extremas, a través de un programa informático que determina la cantidad exacta de agua que se suministra a cada planta. También, esta instalación mide el crecimiento y la capacidad de las plantas para no marchitarse mediante el análisis de imágenes digitales estereoscópicas, que son tomadas en forma automática con una frecuencia de hasta una hora.

Además, si bien este proyecto fue pensado para analizar el comportamiento de la planta de soja, el robot puede ser perfectamente compatible para el trabajo fenotípico de otras plantas que también padezcan condiciones de estrés hídrico.

La utilización de técnicas de procesamiento digital de imágenes para la detección de las características morfológicas de la planta pretende convertirse en un paso significativo para

la optimización de los resultados en los estudios de esta área.

1.2. Procesamiento Digital de Imágenes.

El procesamiento digital de imágenes es un área que viene evolucionando continuamente debido a los avances tecnológicos en imagen digital y por el crecimiento de los procesadores y dispositivos de almacenamiento masivo. Este se basa principalmente en la extracción de información útil a partir de las imágenes.

Algunas de las ventajas de trabajar con imágenes digitales son el bajo costo del procesamiento, la velocidad en la toma de datos y la posibilidad de trasladar la información obtenida directamente a un ordenador pudiendo ser reprocesadas en cualquier otra ocasión.

En este proyecto se plantea procesar estas imágenes con el fin de segmentar, medir y cuantificar el follaje de las plantas de soja mediante la utilización de filtros basados en el diagrama cromático.

La segmentación automática de imágenes constituye un problema significativo dentro del Procesamiento Digital de Imágenes (PDI). Es uno de los elementos más importantes de cualquier sistema automatizado de visión; es el primer nivel de la tarea de entendimiento de la imagen y afecta severamente al proceso posterior de interpretación de la imagen, proporcionando estructuras útiles tales como regiones y bordes.

Debido a las limitaciones de los equipos de captura y procesamiento con respecto a los altos tiempos y costos que conllevaban, el uso de las imágenes en niveles de grises era más beneficioso con respecto a las imágenes en color, perdiéndose información útil. El crecimiento en las capacidades de almacenamiento y poder de cálculo junto con la modernización de sistemas de bajo costo de captura e impresión de imágenes han provocado un interés creciente en la utilización de imágenes en color en el desarrollo de muchas aplicaciones.

Los espacios de color proporcionan un método para especificar, ordenar y manipular colores. En la actualidad existen numerosos espacios de color. La correcta elección de un espacio de representación para el color es hoy en día un desafío para la comunidad científica que investiga el procesamiento y análisis de imágenes en color.

La gran mayoría de los espacios de color se han desarrollado para aplicaciones específicas, aunque todos parten de un mismo concepto: la teoría tricromática de colores primarios rojo, verde y azul.

Muchos de los espacios de color creados para aplicaciones gráficas de los ordenadores son inadecuados para el tratamiento cuantitativo de las imágenes.

Una representación conveniente, debe estar basada en distancias o normas para los vectores de los puntos del espacio de representación y además proporcionar independencia entre las componentes cromática y acromática.

Los filtros color han sido utilizados anteriormente para la medición de biomasa, y han demostrado ser en muchas aplicaciones, más eficientes que las técnicas de segmentación en niveles de gris.

Complementariamente, la utilización de otras técnicas del Procesamiento Digital de Imágenes, permitirá la medición y cuantificación del follaje de las plantas.

Por todo lo expuesto se considera importante investigar nuevas técnicas de segmentación automática en espacios de color que contribuyan a segmentar imágenes.

1.3. Objetivos.

El objetivo general de este proyecto es desarrollar e implementar un software de procesamiento digital de imágenes en Color que permita segmentar estructuras para su posterior cuantificación.

Los objetivos específicos se mencionan a continuación:

Desarrollar los algoritmos correspondientes a las técnicas de procesamiento digital de imágenes (pre-procesamiento y segmentación) e implementarlos en computadora a través de Matlab.

Aplicar las técnicas desarrolladas a las imágenes de plantas de soja suministradas por el INTA Balcarce.

Analizar los resultados y compararlos con los obtenidos con métodos tradicionales.

CAPÍTULO II

TÉCNICAS DE PDI

Capítulo II

TÉCNICAS DE PDI

2.1. Introducción.

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información. Es decir, lo que se busca con dicho procesamiento es mejorar la imagen con el fin de hacerla más adecuada que la original para una determinada aplicación o procesamiento posterior. Por lo tanto, dependiendo del problema específico a resolver se utilizan diferentes técnicas.

Los métodos de mejoramiento de imagen se pueden clasificar en dos grandes grupos: métodos en el dominio espacial y métodos en el dominio de la frecuencia. Los primeros se basan en manipulaciones directas sobre los píxeles de la imagen, mientras que los segundos se basan en modificar la transformada de Fourier de la imagen.

2.2. Planteamiento General

Una imagen se puede describir por medio de una función $f:U \rightarrow [0,255]$, donde $U \subset \mathbb{Z}^2$ es un subconjunto del plano discreto. El conjunto U se denomina el soporte de la imagen y el conjunto de valores de f se denomina el conjunto de valores de la imagen.

La imagen puede ser representada como una matriz, es decir: $M = (a_{ij}) \in [0,255]$ donde $a_{ij} = f(i,j)$, $\forall (i,j) \in U$. Cada a_{ij} indica el nivel de gris de la imagen en el punto (i,j) , el cual se denomina píxel. Por lo tanto, la imagen queda unívocamente determinada por la matriz M .

La figura 2.1 muestra el gráfico de la función imagen. En ella se puede observar que la altura indica la intensidad de gris de cada píxel.

El número m de filas y el número n de columnas de la matriz M representan la resolución espacial de la imagen. Ésta establece la dimensión de la imagen. Por lo tanto, mayor resolución implica una percepción mayor de los detalles de la imagen.



Imagen original

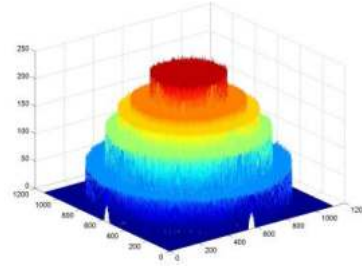


Gráfico de la función imagen

Figura 2.1: representación de una imagen como función

Las imágenes comúnmente manipuladas, son binarias, en niveles de gris o color. Las imágenes binarias son aquellas cuyo conjunto de valores está compuesto por 0 y 1, donde 0 representa el negro y 1 representa el blanco. El conjunto de valores de las imágenes en niveles de gris está compuesto por enteros que varían entre 0 y 255, indicando la intensidad de gris de cada píxel.

Uno de los formatos más utilizados de representación de imágenes color es el sistema RGB. La imagen está formada por tres matrices de m filas por n columnas, donde cada una de ellas representa las componentes R (Rojo), G (Verde) y B (Azul). De esta manera, cada píxel de la imagen tiene tres componentes compuestos por enteros valuados entre 0 y 255. Los diferentes colores se obtienen en forma aditiva, combinando distintas proporciones de rojo, verde y azul. Así el valor 0 para las tres componentes $[0,0,0]$ representa el negro y la combinación $[255,255,255]$ corresponde al blanco. Si una de las componentes es 255 y las otras dos son nulas, se obtienen los colores puros rojo, verde y azul. Si una imagen tiene todos sus píxeles con las tres componentes iguales, da como resultado una imagen en niveles de gris. En la figura 2.2 se muestran ejemplos de estos tipos de imágenes.



Imagen Binaria



Imagen en niveles de gris



Imagen color

Figura 2.2: Ejemplos de imágenes según su composición.

Dadas dos imágenes M_1 y M_2 es posible implementar operaciones *aritméticas* y *lógicas* a nivel de píxeles entre ellas. Si el operador Δ indica una operación definida, entonces podemos construir la siguiente asociación:

$$MR(i, j) = M_1(i, j) \Delta M_2(i, j) \quad (2.1)$$

Esto indica que en la posición (i, j) de la imagen resultado MR se asigna el valor que arroja la operación Δ al operar los píxeles en la posición (i, j) de las imágenes base. A continuación, se describe el procesamiento según el dominio espacial o frecuencial.

2.3. Métodos de procesamiento en el dominio espacial.

En el procesamiento digital de imágenes el conjunto de píxeles que constituyen una imagen es denominado dominio espacial. Los métodos en el dominio espacial se basan en técnicas para manipular dichos píxeles.

Pueden reflejarse las funciones de procesamiento de la imagen en el dominio espacial como:

$$g(x, y) = T[f(x, y)] \quad (2.2)$$

donde se define $f(x, y)$ como la imagen de entrada, $g(x, y)$ como la imagen procesada y T es un operador que actúa sobre f , definido en algún entorno del píxel (x, y) . El operador T puede también operar sobre un conjunto de imágenes de entrada.

La máscara, de geometría cuadrada o rectangular, centrada en el píxel (x, y) determina el entorno sobre el cual se aplica la función. Un entorno de dimensión 3×3 puede observarse en la siguiente figura:

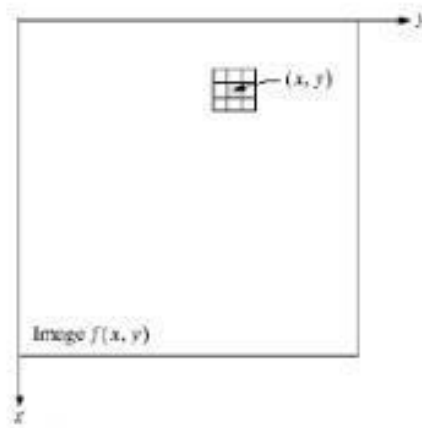


Figura 2.3: ejemplo de procesamiento espacial con un entorno de dimensión 3×3

Resumidamente, se sitúa el centro de la máscara sobre cada píxel modificándolo, utilizando los píxeles de alrededor y sus respectivos valores de ponderación. Una vez calculado el nuevo valor, la máscara se desplaza hacia el siguiente píxel y se realiza la misma operación para obtener la imagen de salida g .

Si por ejemplo se toma un entorno de 1×1 , sólo el valor de f definiría el valor de g en el

punto (x,y) . De este modo T se convierte en una función de transformación del nivel de gris de la forma:

$$s=T(r) \tag{2.3}$$

donde las variables r y s indican el nivel de gris de $f(x,y)$ y $g(x,y)$ en cada punto (x,y) respectivamente.

A continuación, se detallan dos ejemplos simples mostrados en la siguiente figura:

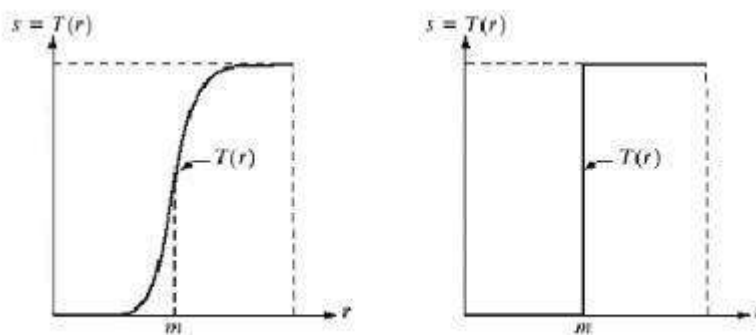


Figura 2.4

En la figura de la izquierda se puede observar que los niveles de gris por encima del umbral m aumentan su iluminación mientras que los niveles de gris que están por debajo se oscurecen, consiguiendo de esta forma una imagen de mayor contraste que la imagen original.

En la figura de la derecha se puede observar el caso límite en el cual la transformación produce una imagen binaria. Las técnicas de esta categoría se conocen a menudo con el nombre de “procesamiento puntual”, debido a que la mejora de cada punto de una imagen depende sólo del nivel de gris en ese punto.

Sin embargo, independientemente de la aplicación específica, la idea general consiste en determinar g en un punto (x,y) a partir de los valores de f en un entorno predefinido de (x,y) . Las técnicas de mejora basadas en este tipo de aproximación se conocen a menudo como procesamiento local o espacial por máscara. Esto se debe a que se modifica el valor de cada píxel utilizando los valores de los píxeles que lo rodean.

2.4. Métodos en el dominio de la frecuencia.

Los métodos en el dominio de la frecuencia se basan en el empleo del teorema de convolución [1]. Sea $g(x,y)$ una imagen formada por la convolución de una imagen $f(x,y)$ y un operador lineal invariante de posición $h(x,y)$, es decir:

$$g(x,y) = h(x,y) * f(x,y) \quad (2.4)$$

Entonces, por el teorema de convolución, se cumple la siguiente relación en el dominio de la frecuencia:

$$G(u,v) = H(u,v)F(u,v) \quad (2.5)$$

Donde G , H y F son las Transformadas bidimensionales de Fourier de las funciones g , h y f respectivamente.

La ecuación anterior describe un proceso espacial que es análogo al empleo de máscaras expuesto anteriormente, pero en el campo transformado. Normalmente, en la terminología de la teoría de Sistemas Lineales, la función $H(u,v)$ se denomina función Transferencia. La misma se selecciona de forma tal que la imagen G resultante resalte determinadas características. Por ejemplo, el diseño de un filtro pasabajos o pasaltos en frecuencia, para obtener una imagen suavizada o una imagen con bordes resaltados respectivamente.

En el ejemplo mostrado en la figura 2.5., se aplica un filtro pasaltos en el campo transformado con el fin de resaltar bordes.

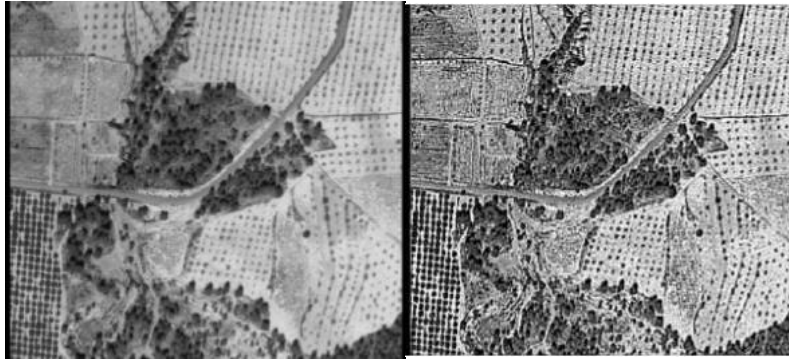


Figura 2.5: Ejemplo de filtro pasaltos.

La incorporación de la Transformada bidimensional de Fourier en el Procesamiento Digital de Imágenes juega un rol muy importante en la mejora, restauración, codificación y descripción de las mismas.

2.5. Morfología Matemática

El procesamiento morfológico de imágenes es una rama del PDI que se ha desarrollado y estudiado ampliamente en los últimos años. Desde una perspectiva científica, la palabra Morfología se refiere al estudio de las formas y estructuras que la materia puede tomar [2]. En procesamiento de imágenes, morfología es el nombre de una metodología específica para analizar las estructuras geométricas dentro de una imagen [2-3].

En los años sesenta dos investigadores de Paris School of Mines, en Fontainebleau, Georges Matheron y Jean Serra trabajaron en problemas de mineralogía y petrografía. Su principal objetivo era caracterizar propiedades físicas de ciertos materiales, como la permeabilidad de medios porosos, examinando su estructura geométrica. Así fue cómo surgió la Morfología Matemática (MM), teoría basada en conceptos de Geometría, Álgebra, Topología y Teoría de Conjuntos [3].

El objetivo principal de la MM es extraer información relativa a la geometría y topología de un conjunto desconocido presente en la imagen. La clave de esta metodología está en el “*elemento estructurante*”, un conjunto completamente definido y de geometría conocida, que es comparado, a partir de una transformación, con el conjunto desconocido de la imagen. La forma y tamaño del elemento estructurante permiten testear y cuantificar de qué manera el elemento estructurante “*está, o no está contenido*” en la imagen [4].

Una de las ventajas de la MM es su simplicidad de implementación y sus pilares son sus dos operaciones básicas, la erosión y la dilatación, a partir de las cuales, por composición,

es posible construir nuevos operadores, lo que hace que la MM se destaque de otras técnicas de procesamiento de imágenes donde, en la mayoría de los casos, las implementaciones no hacen uso de las herramientas ya existentes [5].

Existen tres tipos de MM: morfología matemática binaria que se aplica sobre imágenes binarias, morfología en niveles de gris que se aplica en imágenes con niveles de gris y la morfología en colores.

Debido a que en este trabajo sólo se utiliza la morfología matemática binaria, a continuación se realiza una descripción de sus principios y operaciones.

2.6. Morfología Matemática Binaria

Cómo se indicó anteriormente, la idea central de la Morfología Matemática es examinar las estructuras geométricas de una imagen por superposición con pequeños patrones localizados (denominados elementos estructurantes) en distintas partes de la misma.

Un elemento estructurante es un subconjunto de \mathbb{R}^2 , cuyo tamaño y forma se escoge a priori de acuerdo con la morfología del conjunto sobre el que se va a interactuar y de acuerdo a la extracción de formas que se desean obtener. En la figura 2.6 se muestran ejemplos básicos de elementos estructurantes utilizados en la práctica.



Figura 2.6: Formas básicas de elementos estructurantes planos.

Si B denota un elemento estructurante, \check{B} denotará el conjunto traspuesto, esto es:

$$\check{B} = \{-b : b \in B\} \quad (2.6)$$

\check{B} se obtiene al rotar B 180° sobre el origen O . El origen O es llamado el centro del elemento estructurante B .

B_x denotará la traslación de B por x :

$$B_x = \{b + x : b \in B\} \quad (2.7)$$

Las operaciones u operadores morfológicos erosión y dilatación constituyen la base de esta teoría. La erosión queda completamente sustentada por la definición de resta de Minkowski mientras que la suma de Minkowski sustenta la definición de la dilatación [6].

Al igual que en la bibliografía, en esta tesis se definen en primer término los operadores morfológicos para imágenes binarias y luego se extienden a imágenes en niveles de gris.

Sean X, B dos subconjuntos de \mathbb{Z}^2 , la **erosión** de X por el elemento estructurante B , denotada por $\varepsilon_B(X)$, es el conjunto de puntos $x \in \mathbb{Z}^2$ tal que el conjunto B_x está incluido en X . Formalmente:

$$\varepsilon_B(X) = \{x \in \mathbb{Z}^2 : B_x \subseteq X\} \quad (2.8)$$

Informalmente, la operación erosión consiste en hacer decrecer al conjunto X a través de un proceso controlado de eliminación de elementos, tomando como referencia el elemento estructurante B . El tamaño y forma final del conjunto erosionado dependerá fuertemente del tamaño y forma del elemento estructurante [7]

Como se mencionó anteriormente, la *erosión* queda sustentada a partir de la *resta de Minkowsky*.

Dados dos conjuntos $X, B \subset \mathbb{Z}^2$, se define la **resta de Minkowsky** $X \ominus B$ de X por B como:

$$X \ominus B = \bigcap_{b \in B} X_b \quad (2.9)$$

La relación entre la *erosión* y la *resta de Minkowsky* está dada por la siguiente igualdad:

$$\varepsilon_B(X) = X \ominus \check{B} = \left\{ x \in \mathbb{Z}^2 : \forall b \in \check{B}, x - b \in X \right\} \quad (2.10)$$

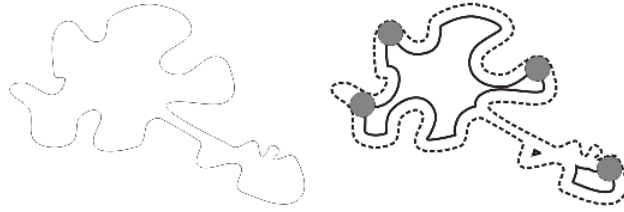


Figura 2.7: Ejemplo de erosión en una imagen binaria euclídea.

En la figura 2.7 se presenta un ejemplo sencillo de una erosión de una imagen con un elemento estructurante circular. El área encerrada por la línea a trazo continuo constituye la erosión de X por B y la línea punteada indica el conjunto original X tomado como referencia.

Sean $X, B \subset \mathbb{R}^2$, la **dilatación** de X por un elemento estructurante B , denotada por $\delta_B(X)$, es el conjunto de puntos $x \in \mathbb{R}^2$ tal que el conjunto B_x tiene intersección no vacía con el conjunto X . Formalmente:

$$\delta_B(X) = \{x \in \mathbb{R}^2 : B_x \cap X \neq \emptyset\} \quad (2.11)$$

En forma coloquial, dilatar la imagen X por el elemento estructurante B consiste en eliminar del fondo todos los puntos x para los cuales el conjunto B_x no esté incluido, o de forma equivale asignar a la imagen dilatada todos los puntos x tales que B_x intercepte a la imagen.

Al igual que con la erosión, la dilatación es sustentada por la *suma de Minkowsky*.

Dados dos conjuntos $X, B \subset \mathbb{R}^n$, se define la *suma de Minkowsky* $X \oplus B$ de X por B como:

$$X \oplus B = \bigcup_{b \in B} X_b \quad (2.12)$$

La relación entre la *dilatación* y la *suma de Minkowsky* está dada por la siguiente igualdad:

$$\delta_B(X) = X \oplus \check{B} = \left\{ x + b : x \in X, b \in \check{B} \right\} \quad (2.13)$$

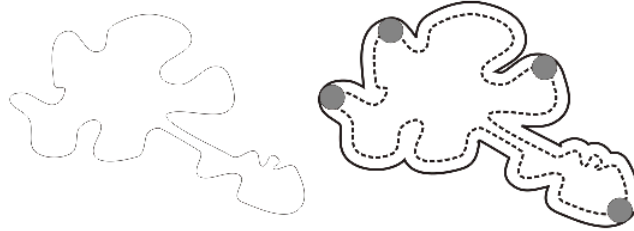


Figura 2.8: Ejemplo de dilatación en una imagen binaria euclídea.

La figura 2.8 muestra un ejemplo de una dilatación de una imagen con un elemento estructurante circular. La zona encerrada por la línea a trazo completo constituye la dilatación de X por B mientras que la línea punteada indica el conjunto original que se toma como referencia.

De los ejemplos de erosión y dilatación mostrados en las figuras 2.7 y 2.8, es claro que el operador erosión “*contrae*” el conjunto, mientras que la dilatación lo “*expande*”.

A través de la dilatación, diferentes componentes de un conjunto se pueden conectar y algunos agujeros se pueden llenar, mientras que a través de la erosión, los huecos se agrandan y algunos objetos pueden desaparecer.

De la combinación de los operadores erosión y dilatación, se derivan nuevas transformaciones morfológicas.

Generalmente, la erosión $X \rightarrow \varepsilon_B(X)$ y la dilatación $X \rightarrow \delta_B(X)$ son operaciones que no admiten inversas, por lo tanto, no hay manera de determinar el origen X desde las imágenes $\varepsilon_B(X)$ o $\delta_B(X)$. Dada esta característica se las puede combinar para generar nuevos operadores.

A continuación se definen las operaciones de apertura y cierre, conocidas en morfología matemática como filtros básicos a partir de los cuales se desarrollan otros más complejos.

Sean $X, B \subset \mathbb{R}^2$, la **apertura (openig)** de X por el elemento estructurante B se define como:

$$\gamma_B(X) = \delta_B(\varepsilon_B(X)) \quad (2.14)$$

En la figura 2.9 se ilustra el efecto de una apertura de un conjunto $X \subset \mathbb{R}^2$ por un elemento estructurante en forma de disco.

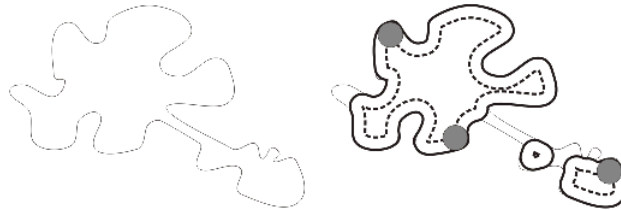


Figura 2.9: Ejemplo de apertura en una imagen binaria euclídea.

Sean $X, B \subset \mathbb{R}^2$, el **cierre (closing)** de X por el elemento estructurante B está definido por:

$$\phi_B(X) = \varepsilon_B(\delta_B(X)) \quad (2.15)$$

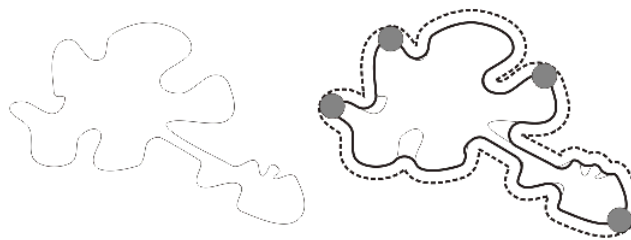


Figura 2.10: Ejemplo de cierre en una imagen binaria euclídea.

En la figura 2.10 se presenta un ejemplo, en el cual a una imagen euclídea se le aplica una operación de cierre con un elemento estructurante en forma de disco.

La apertura y cierre tienen un significado morfológico más fuerte que la erosión y la dilatación, pues permiten filtrar una imagen.

Una operación de suma importancia en morfología matemática es la reconstrucción tanto para imágenes binarias como para imágenes en niveles de gris [8].

Sean X (máscara) e Y (marca) dos imágenes binarias definidas en el mismo dominio D tal que $Y \subseteq X$, esto es, $\forall p \in D, Y(p)=1 \Rightarrow X(p)=1$. Si X_1, X_2, \dots, X_n son las componentes conectadas de X , la **reconstrucción** binaria $\rho_X(Y)$ de la máscara X a partir de la marca Y se define como la unión de todas las componentes conectadas X_k de X que

contienen al menos un punto de Y , esto es:

$$\rho_X(Y) = \bigcup_{x \in (X_k \dots Y \neq \emptyset)} X_k \quad (2.16)$$

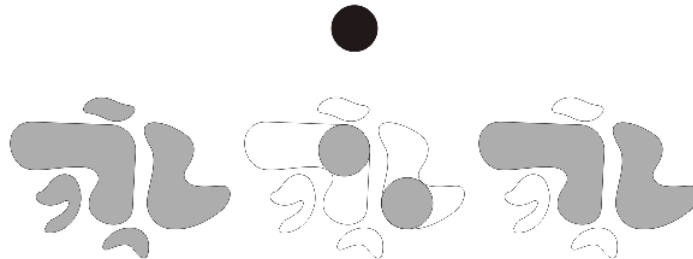


Figura 2.11: Ejemplo de apertura por reconstrucción en una imagen binaria euclídea.

Una aplicación de la reconstrucción binaria es la de filtrar una imagen. Un caso especial dentro de estas transformaciones, es la denominada apertura por reconstrucción.

En imágenes binarias, la apertura por un elemento estructurante es utilizada usualmente para filtrar las componentes conectadas de una imagen que no contienen al elemento estructurante [9]. Sin embargo, en algunos casos se desea filtrar todas las componentes conectadas que no contienen al elemento estructurante y dejar sin modificaciones las restantes. El resultado de esta transformación recibe el nombre de *aperturas por reconstrucción* [10].

La **apertura binaria por reconstrucción** $\gamma_B^{rec}(X)$ de la imagen X por el elemento estructurante B consiste en una apertura binaria de X por B seguida de la reconstrucción de X a partir de la imagen obtenida después de aplicarle el operador apertura $\gamma_B(X)$, esto es:

$$\gamma_B^{rec}(X) = \rho_X(\gamma_B(X)) \quad (2.17)$$

Este operador constituye uno de los filtros morfológicos más interesantes en el tratamiento de imágenes binarias [11].

Mediante la combinación de diferentes técnicas, la MM brinda herramientas adicionales que permiten extraer en imágenes binarias regiones conectadas con áreas inferiores a un

determinado valor umbral λ .

Sea $C_x(A)$ la *región conectada* de la imagen A que contiene al elemento $x \in A$, definida por:

$$C_x(A) = \rho_A(X, B) \quad (2.18)$$

donde el elemento estructurante B define la conectividad y $X = \{x\}$.

El operador morfológico *areaopening*, notado por γ_λ^{area} , de parámetro λ sobre la imagen A queda definido como:

$$\gamma_\lambda^{area}(A) = \{x \in A / Area(C_x(A)) \geq \lambda\} \quad (2.19)$$

El operador *areaopening* elimina aquellas regiones conectadas de la imagen con área menor a λ .

El *areaclosing*, notado por ϕ_λ^{area} , de parámetro λ sobre la imagen A queda definido como:

$$\phi_\lambda^{area}(A) = [\gamma_\lambda^{area}(A^c)]^c \quad (2.20)$$

donde A^c es el complemento de la imagen. El operador *areaclosing* elimina aquellas componentes conectadas con el fondo con un área menor a λ .

Mediante la extensión de los operadores básicos a imágenes en niveles de gris surge la MM en niveles de gris.

2.7. Detección de bordes.

El procedimiento más usual en la detección de discontinuidades es la detección de bordes. Podemos definir a un borde como el límite entre dos regiones con nivel de gris relativamente diferente. Si las regiones de interés son suficientemente homogéneas la transición entre dichas regiones se puede determinar empleando exclusivamente las discontinuidades en el nivel de gris.

El principio básico de un detector de bordes es el cálculo de un operador local de derivación. En la Figura 2.12 se puede ver este concepto. En la parte izquierda se puede ver una

imagen de una banda clara sobre un fondo oscuro, el perfil a lo largo de una línea horizontal y la primera y segunda derivada de dicho perfil. Se puede observar que el perfil del borde se ha modelado como una discontinuidad suave. Esto tiene en cuenta el hecho de que en las imágenes reales los bordes están ligeramente desenfocados.

Como se puede observar en la Figura 2.12 la primera derivada es positiva para cambio a nivel de gris más claro, negativa en caso contrario y cero en aquellas zonas con nivel de gris uniforme. La segunda derivada presenta valor positivo en la zona oscura de cada borde, valor negativo en la zona clara de cada borde y valor cero en las zonas de valor de gris constante y justo en la posición de los bordes. El valor de la magnitud de la primera derivada sirve para detectar la presencia de bordes, mientras que el signo de la segunda derivada indica si el pixel pertenece a la zona clara o a la zona oscura. Además, la segunda derivada presenta siempre un cruce por cero en el punto medio de la transición. Esto puede ser muy útil para localizar bordes en una imagen.

La interpretación anteriormente explicitada se refiere a perfiles unidimensionales. Sin embargo, para perfiles bidimensionales, únicamente se deberá definir el perfil en la dirección perpendicular a la dirección del borde para que lo mencionado anteriormente siga siendo válido. La primera derivada en cualquier punto de la imagen estará dada por la magnitud del **gradiente**, mientras que la segunda derivada por el operador **Laplaciano**.

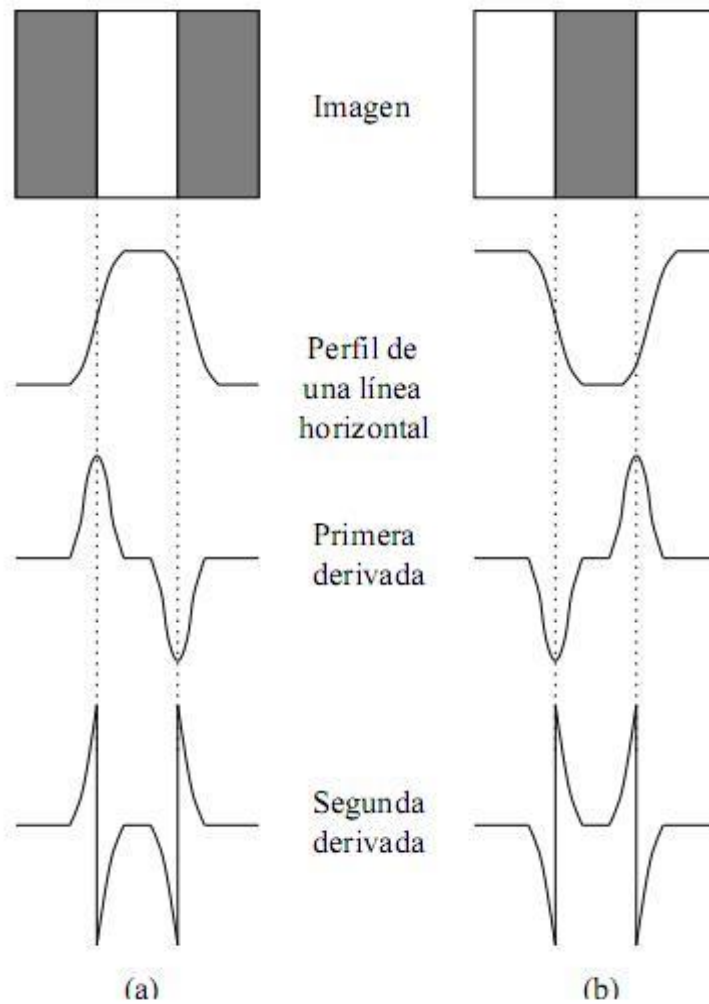


Figura 2.12. Detección de bordes empleando operadores de derivación. La segunda derivada tiene un cruce por cero en la posición de cada borde.

2.8. Gradiente Morfológico

Una forma alternativa para la obtención de los bordes de objetos en una imagen es la utilización del gradiente morfológico. En morfología matemática, existen varias implementaciones digitales del gradiente. Las más usadas son el gradiente morfológico por erosión, el gradiente morfológico por dilatación y el gradiente morfológico por erosión y dilatación.

El gradiente morfológico $gradm_\varepsilon(f, b)$ de una función f por un elemento estructurante b a partir de la erosión es:

$$gradm_\varepsilon(f, b) = f - \varepsilon(f, b) \quad (2.23)$$

El gradiente por erosión en niveles de gris detecta bordes en las posiciones de los niveles de gris más elevados en los bordes.

El gradiente morfológico $gradm_\delta(f, b)$ de una función f por un elemento estructurante b a partir de la dilatación es:

$$gradm_\delta(f, b) = \delta(f, b) - f \quad (2.24)$$

El gradiente por dilatación en niveles de gris detecta bordes en las posiciones de los niveles de gris más bajos en los bordes.

El tercer tipo de gradiente morfológico mencionado, consiste en usar en forma simultánea los operadores de erosión y dilatación. Formalmente, el gradiente morfológico $gradm_{\varepsilon, \delta}(f, b)$ de una función f por un elemento estructurante b a partir de los operadores dilatación y erosión es:

$$gradm_{\varepsilon, \delta}(f, b) = \delta(f, b) - \varepsilon(f, b) \quad (2.25)$$

El gradiente por dilatación y erosión en niveles de gris agrupa los resultados de los gradientes por erosión y dilatación. Por lo tanto, podemos observar que los operadores del gradiente $gradm_\varepsilon(f, b)$ presentan contornos más finos que el operador $gradm_{\varepsilon, \delta}(f, b)$.

2.9. Procesamiento de imágenes en Espacios Color

El procesamiento de imágenes color utiliza técnicas específicas dependiendo del espacio color en el cual se representa y procesa la imagen. Debido a que las imágenes a procesar en esta tesis son imágenes color, y que su procesamiento es la parte más importante en el método propuesto, en los siguientes capítulos se realiza una descripción detallada de los espacios color y del filtrado para este tipo de imágenes.

CAPÍTULO III

SEGMENTACIÓN DE IMÁGENES

Capítulo III

SEGMENTACIÓN DE IMÁGENES

3.1. Introducción.

La habilidad de detectar objetos con características determinadas dentro de una imagen constituye un aspecto fundamental para la automatización de diversos tipos de aplicaciones. Estos tipos de procedimientos, conocidos como segmentación de imágenes, permiten la descomposición de una imagen en regiones de interés, según cada aplicación específica [12]. Esta etapa es de suma importancia en cualquier sistema automatizado de visión, ya que es el primer paso en la tarea de entendimiento de la imagen y afecta severamente al proceso posterior de interpretación de la misma, proporcionando estructuras útiles tales como regiones y/o bordes.

La segmentación de imágenes tiene su origen en numerosos estudios psicológicos que indican la preferencia de los seres humanos por agrupar regiones visuales en términos de proximidad, similitud y continuidad, para construir un conjunto de unidades significativas [13]. En cuanto a la unidad significativa que rige la segmentación de imágenes suele corresponder a píxeles, regiones o contornos que muestran o disciernen una similitud en cuanto a intensidad, color, textura, gradiente local, movimiento, entre otros.

La operación de segmentar una imagen para el cerebro humano es tan automática y veloz que no es posible darse cuenta de su complejidad. Esta complejidad se percibe cuando se intenta reproducir el proceso de segmentación por medio de una computadora. Desafortunadamente, realizar esta separación automática en regiones por medio de algoritmos no es nada sencillo y los criterios de agrupamiento varían de un problema a otro, sin haberse logrado aún, imitar la capacidad visual de los seres humanos.

Los expertos (especialistas en el dominio de la aplicación) son capaces de analizar una imagen y reconocer un objeto con mayor facilidad que un algoritmo diseñado para tal fin. La mayor limitación de los algoritmos es la dificultad de transformar el conocimiento global en operaciones locales computacionales. Sin embargo, la ventaja de los algoritmos computacionales radica en que son capaces de reproducir resultados y eliminar discrepancias.

En los últimos años se ha avanzado en la automatización de la segmentación de imágenes. Sin embargo, aunque la mayoría de los métodos desarrollados proporcionan soluciones para algunos pasos del análisis cuantitativo, muchos no pueden ser utilizados en la práctica

porque no tienen en cuenta aspectos importantes como:

- a) La existencia de distintos objetos que tienen o pueden tener brillos similares.
- b) La ausencia de valores absolutos en las unidades de intensidad.
- c) La presencia de determinados ruidos asociados a cada tecnología en la formación de las imágenes.

A pesar de esto, cada nuevo resultado constituye un pequeño paso hacia la revolución dentro de las técnicas de PDI.

Los métodos propuestos en la literatura para dar solución al problema de la segmentación son muy variados dado que no hay una metodología que pueda ser aplicada a cualquier tipo de escena. No obstante, éstos pueden ser agrupados en cuatro clases: técnicas de umbralización, basadas en contornos, basadas en regiones y técnicas híbridas [14][15][16][17].

3.2. Segmentación Manual.

Las segmentaciones manuales se basan en la selección, por parte de un experto, de los píxeles que pertenecen al objeto de interés de forma interactiva, seleccionando píxel por píxel o bien usando herramientas semiautomáticas. Esta segmentación suele ser subjetiva, dependiente del operador y los resultados obtenidos no son siempre repetibles. De todas maneras, una adecuada segmentación manual realizada por especialistas es necesaria para la evaluación de los métodos de clasificación automáticos o semiautomáticos ya que se la utiliza como referencia o “*gold standard*”.

3.3. Segmentación basada en procesamiento de imágenes.

Las técnicas de umbralamiento (thresholding) utilizan generalmente la información del histograma [18]. Su resultado es satisfactorio sólo si el histograma es bimodal. A pesar de que esta técnica es de fácil implementación, depende fuertemente de la elección del umbral (*threshold*).

No existen en la bibliografía trabajos que den cuenta de la aplicación directa de este método en la segmentación de imágenes agrícolas debido a las limitaciones que conlleva la superposición de intensidades de colores y/o grises que corresponden a distintas estructuras.

Los métodos basados en contornos utilizan la hipótesis que las regiones en una imagen son suficientemente homogéneas y, por lo tanto, la transición entre ellas puede ser determinada sólo por las discontinuidades en los niveles de gris [19]. Entre los detectores de contornos más sencillos se pueden mencionar Sobel, Prewitt, Gradiente o Laplaciano [15][20]. Todos ellos tienen como objetivo seleccionar aquellos píxeles de la imagen que delimitan la frontera del objeto de interés. El sustento matemático de la mayoría de las técnicas de detección de bordes es el cálculo de un operador local derivativo.

Otra técnica que se puede incluir dentro de este grupo es el seguimiento de contornos. Esta técnica agrupa en una cadena puntos de la imagen que tienen propiedades comunes, de modo que las regiones que son separadas por esta colección de puntos tengan un cambio significativo en esas propiedades. Los puntos seleccionados de ésta constituyen el contorno de una región [21].

Los métodos basados en regiones se sustentan en la hipótesis que los píxeles que pertenecen a una misma región, tienen características visuales similares (textura, color, forma, etc.). Se destacan la técnica de división y unión (*split and merge*), y la de crecimiento de regiones (*regiongrowing*) a partir de semillas.

La técnica de crecimiento de regiones es una técnica para extraer regiones de la imagen que están conectadas según un criterio definido. Requiere de un valor inicial, denominado semilla (*seedpoint*), elegido manualmente, a partir de allí se establece una propiedad para determinar el conjunto de píxeles que cumplen dicha propiedad y se va etiquetando (*labeling*) hasta que se alcanza el contorno del objeto a identificar [22].

3.4. Segmentación basada en reconocimiento de patrones.

Un paradigma muy utilizado para la segmentación consiste en el agrupamiento o clasificación de los píxeles mediante diversas técnicas. Los píxeles quedan caracterizados por vectores numéricos que los identifican con algún criterio, siendo sus componentes los denominados característicos o descriptores. Estos vectores ingresan al algoritmo de clasificación.

Los característicos, también denominados descriptores, se extraen de las imágenes a segmentar. La selección de un buen conjunto de descriptores es una etapa clave en cualquier proceso de clasificación, en este caso para lograr una segmentación adecuada.

La combinación de dos o más técnicas da lugar a los denominados “modelos híbridos”, de gran aplicación en diversos campos y en particular en la segmentación de imágenes [23].

3.5. Umbralización.

Una de las etapas más importantes dentro del procesamiento es la de umbralización. Como se mencionó anteriormente no existe un método para la selección de un umbral óptimo dependiente de la aplicación. Al tomar como ejemplo el histograma de los niveles de gris de una imagen $f(x,y)$, como el que se muestra en la figura 3.1(a), se observa que dicha imagen está compuesta de objetos claros sobre un fondo oscuro, o viceversa, de tal forma que los niveles de gris están agrupados en dos grupos predominantes. Una posibilidad de separar los objetos del fondo consistiría en seleccionar un umbral T óptimo que separe esos grupos. Entonces, cualquier punto (x,y) para el que se cumpla que:

$$f(x,y) > T \tag{3.1}$$

se lo etiqueta como objeto; en otro caso, como fondo. La figura 3.1(b) muestra el histograma de otra imagen en un caso más general. En este ejemplo el histograma de la imagen está caracterizado por tres grupos dominantes.

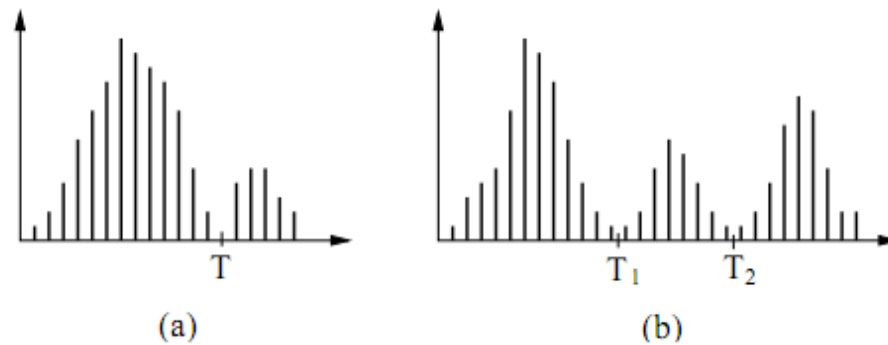


Figura 3.1. Ejemplos de histogramas de niveles de gris que se pueden segmentar con (a) un único umbral y (b) con múltiples umbrales.

Esto ocurrirá, por ejemplo, cuando se tengan dos tipos de objetos claros sobre fondo

oscuro. Se puede utilizar el mismo principio para clasificar cada punto (x, y) .

Si $T_1 < f(x, y) \leq T_2$ entonces se lo etiqueta como primer objeto, si $f(x, y) > T_2$ como segundo objeto y si $f(x, y) \leq T_1$ como fondo. En general, este tipo de clasificación con varios umbrales es menos viable, ya que es más difícil determinar los umbrales que separen de forma efectiva las regiones de interés, especialmente cuando el número de grupos del histograma aumenta. En tal caso es recomendable utilizar umbrales variables.

En general, un método de umbral se puede ver como una operación en la que se hace un test de cada pixel con respecto a una función T de la forma

$$g(x, y) = \begin{cases} 1, & \text{si } f(x, y) > T \\ 0, & \text{si } f(x, y) \leq T \end{cases} \quad (3.2)$$

Entonces los pixeles etiquetados con 1 (o cualquier otro nivel de intensidad conveniente) corresponden a los objetos, mientras que los pixeles etiquetados con 0 corresponden al fondo. En la práctica, el tipo de umbral global dado por la ecuación anterior es útil en ambientes altamente controlados. Una de las áreas donde es posible utilizarlo es en aplicaciones de inspección industrial, donde el control de la iluminación usualmente es posible y, por lo tanto, se puedan obtener histogramas con 2 grupos fáciles de separar.

3.6. Umbral óptimo.

3.6.1. Introducción.

La mayoría de las técnicas de umbralización se basan en estadísticas sobre el histograma unidimensional de una imagen. Para localizar los umbrales se pueden usar procedimientos paramétricos y no paramétricos. En los paramétricos, la distribución de los niveles de gris de una clase de objeto lleva a encontrar los umbrales. En los procedimientos no paramétricos, los umbrales se obtienen de forma óptima de acuerdo a algún criterio. En particular, el método de Otsu, elige el umbral óptimo maximizando la varianza entre clases mediante una búsqueda exhaustiva[1].

Si bien hay diferentes métodos para hallar un umbral, la mayoría de ellos no dan buenos resultados cuando se trabaja con imágenes del mundo real debido a la presencia de ruido, imágenes con histogramas planos o iluminación inadecuada. Por el contrario, el método de Otsu es uno de los mejores métodos de selección de umbral para imágenes del mundo real.

Este método usa una búsqueda exhaustiva para evaluar el criterio que maximiza la varianza entre clases. Sin embargo, a medida que el número de clases de una imagen aumenta, este método necesita mucho más tiempo para seleccionar un umbral multinivel adecuado. La importancia del método de Otsu radica en que es automático, es decir, no necesita supervisión ni información previa de la imagen antes de su procesamiento. Debido a la importancia de este método en el desarrollo de ésta tesis en el siguiente apartado se realiza una explicación más detallada del mismo.

3.6.2. Método de Otsu.

Cómo se definió en capítulos anteriores, una imagen es una función bidimensional de la intensidad del nivel de gris, y contiene N píxeles cuyos niveles de gris se encuentran en el intervalo natural $[0, 255]$. El número de píxeles con nivel de gris i se denota como f_i , y la probabilidad de ocurrencia del nivel de gris i en la imagen está dada por:

$$p_i = \frac{f_i}{N} \quad (3.3)$$

En el caso de la umbralización en dos niveles de una imagen (comúnmente llamada binarización), los píxeles son divididos en dos clases: C_1 , con niveles de gris $[1, t]$; y C_2 , con niveles de gris $[t+1, L]$. Entonces, la distribución de probabilidad de los niveles de gris para las dos clases es:

$$C_1: \frac{p_1}{\omega_1(t)}, \dots, \frac{p_t}{\omega_1(t)} \quad C_2: \frac{p_{t+1}}{\omega_2(t)}, \dots, \frac{p_L}{\omega_2(t)} \quad (3.4)$$

donde

$$\omega_1(t) = \sum_{i=1}^t p_i \quad \omega_2(t) = \sum_{i=t+1}^L p_i \quad (3.5)$$

También, la media para la clase C_1 y la clase C_2 es

$$\mu_1 = \sum_{i=1}^t \frac{i \cdot p_i}{\omega_1(t)} \quad \mu_2 = \sum_{i=t+1}^L \frac{i \cdot p_i}{\omega_2(t)} \quad (3.6)$$

Sea μ_T la intensidad media de toda la imagen. Es fácil demostrar que:

$$\omega_1 \cdot \mu_1 + \omega_2 \cdot \mu_2 = \mu_T \quad \omega_1 + \omega_2 = 1 \quad (3.7)$$

Usando análisis discriminante, Otsu definió la variancia entre clases de una imagen umbralizada como:

$$\sigma_B^2 = \omega_1 \cdot (\mu_1 - \mu_T)^2 + \omega_2 \cdot (\mu_2 - \mu_T)^2 \quad (3.8)$$

Para una umbralización de dos niveles, Otsu verificó que el umbral óptimo t^* se elige de manera que σ_B^2 sea máxima, esto es:

$$t^* = \text{Max}\{\sigma_B^2(t)\} \quad 1 \leq t \leq L \quad (3.9)$$

En la figura 3.2.b) se muestra el resultado de aplicar el algoritmo Otsu por ventanas a la imagen en niveles de gris de la figura 3.2.a). La necesidad de usar un umbralado automático adaptativo (local) puede entenderse observando las siguientes figuras 3.2.c) a 3.2.f), donde se ha utilizado un umbral fijo para binarizar la imagen. Puede verse que la parte superior ya está saturada, por lo que valores menores de umbral solo empeorarían esta situación, y la parte inferior ya muestra regiones con objetos apenas o no detectados, por lo que valores mayores del umbral solo empeorarían dicha situación. El umbral adaptativo por ventana soluciona este inconveniente.

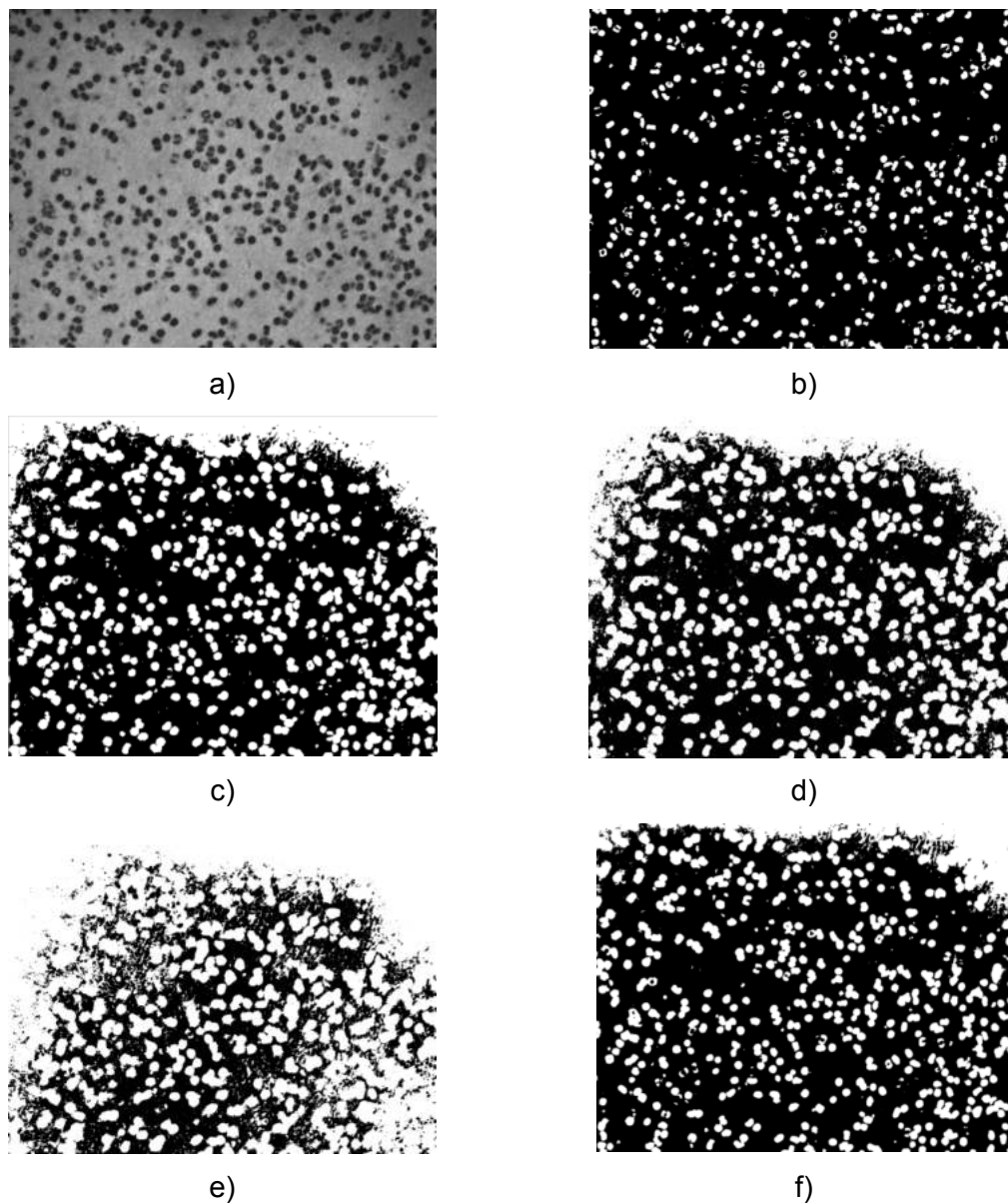


Figura 3.2.: Ejemplos de umbralamiento. a) Imagen original, b) Imagen umbralada mediante algoritmo de Otsu, c) a e) Umbralamiento fijo para distintos valores de umbral, establecido manualmente, f) Umbralamiento de Otsu global

Las técnicas presentadas en este capítulo, conjuntamente con el procesamiento en espacios color que se explican en el próximo capítulo conforman la parte principal del desarrollo de esta tesis para medir en forma automática el área foliar de plantas de soja.

CAPÍTULO IV

ESPACIO COLOR

Capítulo IV

ESPACIO COLOR

4.1. Introducción.

Un espacio de color define un modelo de composición del color. Por lo general un espacio de color lo define una base de N vectores cuya combinación lineal genera todo el espacio de color. Los espacios de color más generales intentan englobar la mayor cantidad posible de los colores visibles por el ojo humano, aunque existen espacios de color que intentan aislar tan solo un subconjunto de ellos.

Existen espacios de color de:

- Una dimensión: escala de grises, escala Jet.
- Dos dimensiones: sub-espacio rg, sub-espacio xy.
- Tres dimensiones: espacio RGB, XYZ, CIELAB, HSV, YCbCr, YUV, YI'Q'.
- Cuatro dimensiones: espacio CMYK.

Los espacios de color de tres dimensiones son los más extendidos y los más utilizados. Por lo tanto, un color se especifica usando tres coordenadas, o atributos, que representan su posición dentro de un espacio de color específico. Estas coordenadas no indican cuál es el color, sino que muestran dónde se encuentra un color dentro de un espacio de color en particular.

Dentro del procesamiento de imágenes, el color es un descriptor sumamente importante y utilizado para la segmentación de imágenes. En aplicaciones relacionadas a las ciencias agrarias el color se utiliza para separar la maleza de la planta en estudio. Esto no es una tarea trivial, dado que existen solapamientos y oclusiones entre las mismas. La idea principal del análisis del color es obtener datos relevantes en algún sistema, para crear patrones que puedan llegar a ser utilizados por otros procesos.

En esta sección se detallarán las principales características de los espacios color que fueron estudiados en esta tesis.

4.2. Sistemas de color.

4.2.1. El sistema CIE-RGB.

La CIE (Comisión Internationale de l'Éclairage) definió en 1931 el sistema CIE RGB. Éste establece un espacio color tricromático (tres colores), cuya base de colores son colores puros en la zona baja, media y alta del espectro visible, o rojo (R), verde (G) y azul (B), respectivamente. Las longitudes de onda (λ) de las radiaciones de partida son: 700 nm para el rojo, 546 nm para el verde y 435.8 nm para el azul.

Los gráficos de las funciones de reconstrucción $R(\lambda)$, $G(\lambda)$ y $B(\lambda)$ se muestran a continuación en la figura 4.1.

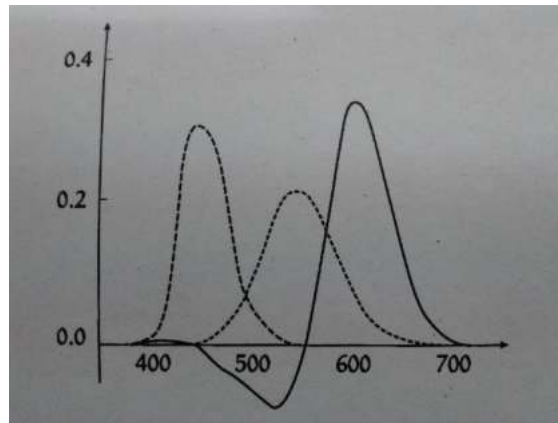


Figura 4.1 – Funciones de reconstrucción de color

Estos valores tabulados permiten obtener la imagen del mapa de colores espectrales en el espacio color:

$$\Phi(\lambda) = (R(\lambda), G(\lambda), B(\lambda)) \quad (4.1)$$

Los dos puntos finales del mapa representan las frecuencias en los límites del espectro visible: la longitud más larga de onda es la roja y la más corta es la azul-violeta. El segmento que une estos puntos contiene varios puntos con matices púrpura, obtenidos de interpolar rojo y azul, por esto se denomina línea púrpura.

El diagrama de cromaticidad del sistema de color CIE-RGB se obtiene proyectando éste en el plano de Maxwell $x+y+z=1$. El borde del diagrama está dado por la proyección radial del mapa $\Phi(\lambda)$ junto con la línea púrpura. La proyección del mapa color se expresa por:

$$C = \frac{1}{R(\lambda) + G(\lambda) + B(\lambda)} (R(\lambda), G(\lambda), B(\lambda)) \quad (4.2)$$

Para obtener una representación bidimensional del diagrama, se debe proyectar ortogonalmente el plano cromático en el plano RG del espacio color; esto significa que se descarta la última coordenada del mapa. El resultado se muestra en la figura 4.2

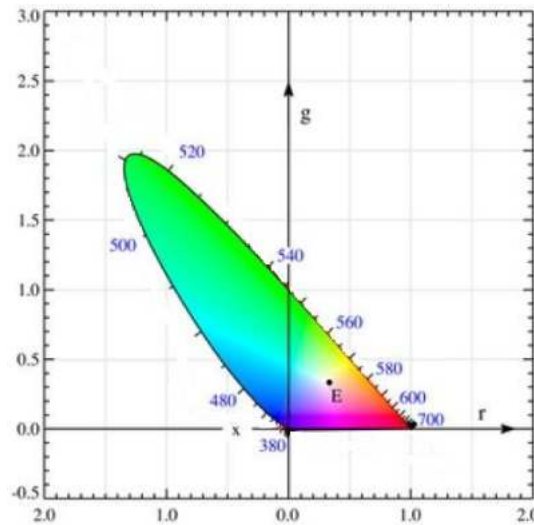


Figura 4.2: diagrama de cromaticidad del sistema de color CIE-RGB

Las unidades de medida a lo largo del sistema CIE-RGB son ajustadas de tal manera que las coordenadas de cromaticidad de igual energía blanca son $(1/3, 1/3, 1/3)$. Esto muestra que la luminancia de un color C con coordenadas (R, G, B) en el sistema CIE-RGB está dado por:

$$L(C) = 0.176R + 0.81G + 0.011B \quad (4.3)$$

El principal problema de esta representación es que la mayoría de los colores tienen una de las coordenadas negativas, lo que dificulta los cálculos colorimétricos y el cálculo de la luminancia partiendo de las componentes tricromáticas.

4.2.2. El sistema CIE-XYZ.

El modelo standard estudiado precedentemente presenta varios inconvenientes. En primer lugar, las bases primarias nos expanden al color sólido. Además, las funciones de reconstrucción toman valores negativos que complican los cálculos de las coordenadas color. Las magnitudes fotométricas no son obtenidas directamente de las coordenadas tridimensionales. Por ejemplo, para calcular la luminancia se debe integrar o utilizar una combinación lineal. Por último, para obtener un punto acromático con coordenadas de cromaticidad $(1/3, 1/3, 1/3)$ es necesario cambiar las escalas de los colores primarios, por lo tanto, la región por debajo del gráfico de la función de reconstrucción color no tiene la misma área (ver figura 4.1).

Debido a estos problemas la CIE estableció, en 1931, un nuevo standard con colores primarios X, Y y Z. Esta designación mejora significativamente la representación de las magnitudes colorimétricas optimizando los cálculos [24].

En este sistema, los primarios satisfacen las siguientes condiciones:

- 1- Todas las componentes XYZ para todos los colores visibles deben ser no negativos.
- 2- Dos de los primarios deben tener luminancia cero.
- 3- Tantos colores espectrales como sean posibles deben tener al menos una componente XYZ cero.

A continuación, se analiza qué significan estas condiciones en términos del diagrama CIE-RGB.

Las funciones de reconstrucción de color para el sistema CIE-RGB toman valores negativos porque los primarios son colores visibles. Eligiendo 3 colores primarios R, G, B que se encuentren en el espectro visible se obtiene un triángulo en el plano de cromaticidad; cualquier color fuera del triángulo necesariamente tiene al menos una coordenada RGB negativa. Para evitar esta situación, se deben elegir colores primarios tales que el triángulo que ellos formen encierre completamente el conjunto de colores visibles, esto es el diagrama de cromaticidad de la figura 4.3. Tales primarios no corresponden al estímulo color físico pero las ventajas computacionales pesaron más que estas desventajas.

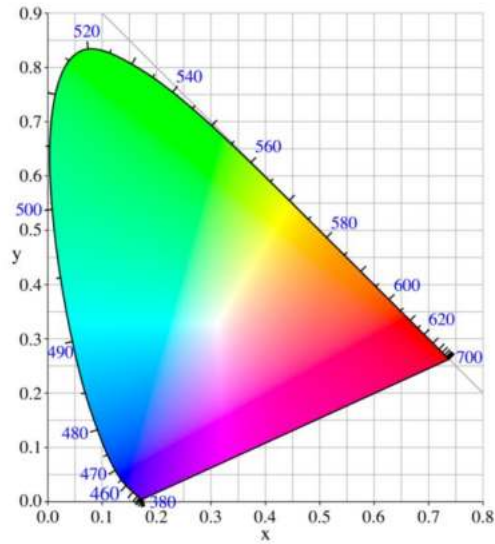


Figura 4.3: diagrama de cromaticidad del sistema de color CIE-XYZ

Para simplificar los cálculos de magnitudes fotométricas, la CIE declaró que dos de los primarios, X y Z deben tener luminancia cero, mientras que la función reconstrucción del color primario Y debe tener la función de eficiencia de luz $V(\lambda)$ en sí misma. Esta condición es consistente con lo anterior ya que los colores primarios no necesitan ser colores visibles, pero sí vectores en nuestro espacio color tridimensional.

Para obtener bases en XYZ que satisfagan la segunda condición, es necesario realizar la descomposición luminancia-crominancia del espacio CIE-RGB, eligiendo vectores X y Z en la componente de crominancia del espacio y el vector Y a lo largo de la componente de luminancia. En otras palabras, los vértices X y Z del triángulo de los primarios deberán encontrarse en la línea de luminancia cero. Por (4.3) el plano de luminancia cero está dado por:

$$0.176R + 0.81G + 0.011B = 0 \quad (4.4)$$

La línea de luminancia cero en el plano RG del diagrama de cromaticidad se obtiene sustituyendo $B=1-R-G$ en la ecuación (4.4), obteniendo:

$$0.066R + 0.70G + 0.11 = 0 \quad (4.5)$$

En esta línea del plano RG se deben colocar los colores primarios X y Z.

Finalmente, para obtener algunos colores espectrales se debe tener al menos alguna componente XYZ cero, por lo tanto, se debe marcar de tal manera que incluya la mayor cantidad de valores del mapa de colores espectrales. El lado XY del triángulo puede ser tomado de la porción del mapa de colores espectrales que va del rojo al amarillo, el cual es siempre directo. El lado YZ del triángulo es elegido arbitrariamente de tal manera que el área del triángulo quede minimizada en términos de coordenadas RGB.

4.2.3. Diagrama cromático.

Las cromaticidades de un color se definen como:

$$t_k \approx \frac{T_k}{T_1 + T_2 + T_3} \quad k = 1, 2, 3 \quad (4.6)$$

Donde T_1 , T_2 y T_3 en la ecuación (4.6) representan las coordenadas color en el espacio tridimensional correspondiente

Se observa que $t_1 + t_2 + t_3 = 1$. Además, solamente dos de las tres coordenadas de cromaticidad son independientes. Por lo tanto, las coordenadas de cromaticidad proyectan el color tridimensional en un plano. Las cromaticidades t_1 , t_2 en forma conjunta representan las componentes de cromaticidad (es decir: saturación y el matiz) de un color. El espacio completo de un color puede ser representado por las coordenadas: (t_1, t_2, Y) en el cual $Y = constante$ es el plano de cromaticidad. El diagrama de cromaticidad representa el subespacio color en el plano de cromaticidad. La figura 4.3 muestra el diagrama para un sistema espectral primario (del CIE).

El diagrama de cromaticidad tiene las siguientes propiedades:

1. La ubicación de todos los puntos que representan el espectro de colores contiene la región de todos los colores visibles.
2. La línea recta que une las coordenadas de cromaticidad del azul (360 nm) y las rojas (780 nm) contiene colores púrpuros y se denomina línea de los púrpuros.
3. La región limitada por las líneas rectas que unen las coordenadas (0,0), (0,1) y (1,0), contiene todos los colores reproducibles por las fuentes primarias. Esta región se denomina gama de colores de fuentes primarias.
4. La referencia de blanco del sistema primario CIE tiene coordenadas de cromaticidad (1/3,1/3). Los colores que se encuentran cercanos a este punto son los colores menos

saturados. Los colores que se encuentran lejanos a este punto son los colores más saturados. Por lo tanto, los colores espectrales y los colores de la línea púrpura son los más saturados.

4.2.4. Transformación entre el sistema CIE-RGB y el sistema CIE-XYZ.

No todos los colores del sistema CIE-XYZ son realizables físicamente. Se pueden obtener magnitudes colorimétricas en este sistema en términos de las del sistema CIE-RGB. Si conociéramos los vectores correspondientes a los colores primarios X, Y, Z, el problema se reduciría a un cambio de base de un espacio vectorial, como se vio en párrafos precedentes. Sin embargo, sólo conocemos las coordenadas de cromaticidad de los colores primarios, por lo tanto, la transformación no es directa. Estos cálculos son importantes, no sólo porque permiten obtener los valores de un color en el sistema XYZ en términos de sus valores en el sistema RGB; sino porque sirven como fundamento para cálculos de cambios de sistemas entre otros sistemas de color.

Para empezar, se deben realizar selecciones y cálculos computacionales basados en las condiciones precedentes que nos permitan obtener las coordenadas de cromaticidad en el sistema CIE-XYZ de los tres colores primarios R, G, B, del sistema CIE-RGB.

Primarios	XYZ cromaticidad		
	x	y	z
R	0.73467	0.26533	0.0
G	0.27376	0.71741	0.00883
B	0.16658	0.00886	0.82456

Estas coordenadas de cromaticidad determinan que tres colores en el espacio color CIE-XYZ, sean:

$$\begin{aligned}
 C_1 &= \rho * (0.73467, & 0.26533, & 0.0) \\
 C_2 &= \gamma * (0.27376, & 0.71741, & 0.00883) \\
 C_3 &= \beta * (0.16658, & 0.00886, & 0.82456)
 \end{aligned}
 \tag{4.7}$$

Donde ρ, γ, β son números reales positivos. Estos tres valores constituyen la base del espacio, por lo tanto el cambio de la matriz de coordenadas debería mapearse con los

valores $R=(1,0,0)$, $G=(0,1,0)$ y $B=(0,0,1)$ a C_1, C_2, C_3 , respectivamente. Esta transformación está dada por:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.73467\rho & 0.27376\gamma & 0.16658\beta \\ 0.26533\rho & 0.71741\gamma & 0.00886\beta \\ 0.0\rho & 0.00883\gamma & 0.82456\beta \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (4.8)$$

Las constantes ρ, γ, β deben determinarse. Para esto requerimos que el color blanco tenga coordenadas $(1/3, 1/3, 1/3)$ en ambos sistemas, esto es,

$$T\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \quad (4.9)$$

Sustituyendo esto en (4.8) se obtiene,

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.73467 & 0.27376 & 0.16658 \\ 0.26533 & 0.71741 & 0.00886 \\ 0.0 & 0.00883 & 0.82456 \end{pmatrix} \begin{pmatrix} \rho \\ \gamma \\ \beta \end{pmatrix} \quad (4.10)$$

Cuya solución es:

$$\rho = 0.666952, \quad \gamma = 1.132407, \quad \beta = 1.200641 \quad (4.11)$$

Sustituyendo estos valores en (4.8) finalmente se obtiene la transformación deseada,

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.489989 & 0.310008 & 0.2 \\ 0.176962 & 0.81240 & 0.010 \\ 0.0 & 0.01 & 0.99 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (4.12)$$

La transformación inversa, que va del sistema XYZ al sistema RGB, está dada por:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 2.3647 & -0.89658 & -0.468083 \\ -0.515155 & 1.426409 & -0.088746 \\ -0.005203 & -0.014407 & 1.0092 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (4.13)$$

Las coordenadas de cromaticidad x, y, z de un color con coordenadas X, Y, Z en el espacio CIE-XYZ, están dadas por

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad z = \frac{Z}{X+Y+Z} \quad (4.14)$$

Sustituyendo los valores de X,Y,Z dados en (4.12), obtenemos inmediatamente la expresión de x,y,z en términos de coordenadas de cromaticidad r,g,b en el sistema CIE-RGB:

$$\begin{aligned} x &= \frac{0.49000r + 0.31000g + 0.2000b}{0.66697r + 1.13240g + 1.20063b} \\ y &= \frac{0.17697r + 0.81240g + 0.01063b}{0.66697r + 1.13240g + 1.20063b} \\ z &= \frac{0.01000 + 0.99000b}{0.66697r + 1.13240g + 1.20063b} \end{aligned} \quad (4.15)$$

Análogamente, usando (4.13) obtenemos r, g, b en función de x, y, z. En particular, las coordenadas de cromaticidad de XYZ primarios en el sistema RGB están dadas por:

Primarios	RGB cromaticidad		
	r	g	b
x	1.2750	-0.2779	0.0029
y	-1.7395	2.7675	-0.0280
z	-0.7431	0.1409	1.6022

La mayoría de los paquetes de software que trabajan con imágenes color utilizan el sistema sRGB. En este caso, se asume que una imagen de 8 bits dispuesta en una pantalla con un buffer de 8 bits por canal se mostrará correctamente. Por esta razón se puede asumir que cualquier imagen de 8 bits está dentro del espacio de color sRGB (en ausencia de cualquier perfil de color incluido en la imagen). De la misma forma, aquellos dispositivos como pantallas LCD, cámaras digitales o impresoras, aunque no producen por naturaleza una curva sRGB, están contruidos con sistemas de circuitos o programas de compensación que al final obedecen este estándar (aunque esto es menos cierto en equipos profesionales). Por esta razón se debe asumir que casi cualquier imagen con 8 bits por canal estará dentro del sRGB.

Considerando que la primera versión del software desarrollado en esta tesis se programó en lenguaje MatLab, el cual utiliza el espacio de representación color sRGB, la matriz de transformación que se utiliza para este caso particular es:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4124564 & 0.3575761 & 0.18044375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{4.16}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

A esta última matriz de transformación la notaremos con la letra M y su inversa M⁻¹.

En el siguiente capítulo se presentará el método de segmentación propuesto utilizando filtros color basados en el diagrama cromático.

CAPÍTULO V

METODO PROPUESTO

Capítulo V

METODO PROPUESTO

5.1. Introducción.

En este capítulo se presenta el algoritmo propuesto en esta tesis para la segmentación y medición del área foliar de una planta de soja de manera automática. Este algoritmo, basado en un filtrado color en el diagrama cromático puede discriminar objetos de color verde del fondo de la imagen. Una vez lograda la segmentación de la planta respecto del fondo se propone filtrar el ruido mediante operadores de la Morfología Matemática. La elección de estos operadores se sustenta en que permiten filtrar el ruido sin modificar la forma original de los objetos de interés. El área del follaje se determinó en base al número de píxeles que componen las hojas segmentadas. En el siguiente capítulo se presentarán las pruebas de validación del algoritmo, que consistieron en aplicar el algoritmo propuesto a la medición del follaje de aproximadamente cincuenta plantas de varios genotipos de soja, en diferentes etapas de crecimiento y bajo distintas condiciones de adquisición, es decir alta variación en los niveles de brillo y sombras.

5.2. Imágenes a procesar

Las imágenes usadas en este trabajo fueron obtenidas de una plataforma automática, que permite fotografiar una gran cantidad de plantas de soja en forma totalmente automática, ubicada en la Unidad Integrada Balcarce, formada por la Facultad de Ciencias Agrarias (FCA) de la Universidad Nacional de Mar del Plata y el Instituto Nacional de Tecnología Agropecuaria (INTA). Las imágenes digitales obtenidas corresponden a vistas superiores y laterales de las plantas con una resolución de 640x480 píxeles, en el modelo de color RGB y formato JPG.

5.2.1. Captura de Imágenes

La utilización de un adecuado protocolo para captura de imágenes, reduce el tiempo y la complejidad de las técnicas de procesamiento digital de imágenes. Es importante destacar que una imagen que no cumpla con los requisitos del protocolo puede provocar una baja tasa de aciertos en la etapa de procesamiento. El protocolo se diseña en base a las características deseables a detectar, a las herramientas

tecnológicas utilizadas y al espacio físico donde se encuentran las plantas. El protocolo de captura de imágenes, que se elaboró en el marco de esta tesis, contempla los siguientes pasos:

Paso 1: Ubicación de la planta.

Las plantas a ser fotografiadas están agrupadas según las condiciones a las que fueron sometidas. Es decir, se agrupan según su exposición a escasez o exceso de agua, luz solar, temperatura y a distintos tipos de tierra.

La planta estará ubicada en una maceta de color blanco, plantada en tierra con una especie de collar blanco en forma de cono alrededor del tallo, como se muestra en la figura 5.1. Dicho cono posee el tamaño suficiente para abarcar todo el volumen de la planta. Esto permite obtener un mayor contraste entre la planta y su entorno. La maceta se identifica con un código escrito en la misma.

Paso 2: Posicionamiento de la cámara fotográfica.

La cámara está adherida a un brazo robotizado que se desplaza planta por planta tomando las imágenes necesarias. La cámara se coloca a una distancia de un metro desde la lente a la base de la maceta. Se desactiva la opción de flash, debido a que dicha luz provoca un exceso de iluminación no deseada.



Figure 5.1: Plataforma robot de ensayo para plantas de soja.

Fijadas las condiciones de adquisición, se presenta a continuación el método propuesto para la segmentación y medición del área foliar de una planta de soja de manera automática.

5.3. Método propuesto

El método propuesto para la medición automática del área foliar se puede resumir en las siguientes cuatro etapas:

1. Filtrado color
2. Binarización
3. Eliminación de ruido
4. Segmentación del follaje y cálculo del área foliar

5.3.1 Filtrado color.

El filtrado color es fundamental y esencial como etapa preliminar de un proceso de segmentación en imágenes color, debido a que éste es un poderoso descriptor que simplifica la identificación de objetos y su extracción de una escena. Sin embargo, este proceso enfrenta una serie de dificultades inherentes a la adquisición de una imagen, como las causadas por el cambio de las condiciones de iluminación y cuando el fondo de la escena contiene superficies y objetos con colores similares al objetivo. Por lo tanto, para este tipo de filtrado es importante seleccionar un modelo de color que separe la información de brillo de la de cromaticidad, en base a la cual se distinguirá el objeto de interés del fondo de la imagen.

Filtro basado en diagrama cromático.

El diseño de este filtro se basa en el diagrama cromático presentado en el Capítulo IV, es decir, para cada píxel de la imagen con sus componentes RGB obtener a través de transformaciones matemáticas lineales sus componentes xy en el plano del diagrama cromático y en base a él decidir si el color del píxel es parte de la planta de soja o no. Nuevamente, si es parte de la planta se deben mantener sus componentes RGB y si no, colocar las componentes RGB de dicho píxel en negro ($R=0, G=0, B=0$). En la actualidad, en el procesamiento de imágenes se utilizan varios espacios de color, definidos a partir de diferentes colores primarios. Las transformaciones entre espacios de color involucran simplemente operaciones matriciales y transformaciones lineales. Como se explicitó en el Capítulo IV, la matriz elegida para este trabajo que permite la transformación del espacio de color XYZ al $sRGB$ es la siguiente:

$$M = \begin{pmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{pmatrix} \quad (5.1)$$

Por lo tanto, dado que el mapa de color de las imágenes leídas por la rutina desarrollada en Matlab es *sRGB*, se debe previamente multiplicar al vector mapa por la inversa de la matriz *M* y obtener de esa manera un mapa en coordenadas *XYZ*.

Si se conviene en reducir todos los colores a la misma intensidad luminosa, entonces los colores pueden ser representados por sólo dos coordenadas. Éste es el caso de las coordenadas del diagrama cromático. Cuando se utiliza el formato de color dado por las coordenadas *CIE-XYZ*, se utilizan las coordenadas de cromaticidad *xy-CIE* que están definidas por las siguientes ecuaciones:

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad (5.2)$$

El código Matlab correspondiente a esta etapa se presenta como función *mapafiltradototal*.

```
function [mapfilttotal] = mapafiltradototal (map)
mr=map(:, :,1); %matriz rojo de mxn
mg=map(:, :,2); %matriz verde de mxn
mb=map(:, :,3); %matriz azul de mxn
mr1=double(mr(:)); %vector columna de (mxn)x1 de la matriz rojo
mg1=double(mg(:)); %vector columna de (mxn)x1 de la matriz verde
mb1=double(mb(:)); %vector columna de (mxn)x1 de la matriz azul
vc=[mr1 mg1 mb1]; %da como resultado una matriz (mxn)x3 que es el
vector color de "map"
vf=vc'; %matriz 3x(mxn)
M=[3.240479 -1.537150 -0.498535;-0.969256 1.875992 0.041556;0.055648
-0.204043 1.057311]; %matriz M de transformación
[f,c]=size(vf);
vectorxyz=ones(f,c); %predefino el tamaño del vector xyz
for c1=1:c
    vectorxyz(:,c1)=M\vf(:,c1); %da como resultado una matriz 3x(mxn)
con la transformación M-1
end
[f,c]=size(vectorxyz);
coc=sum(vectorxyz); %devuelve un vector fila 1xc con la suma de cada
columna (X+Y+Z)
```

```

vecxy=ones(2,c); %predefino el tamaño del vector xy
for c1=1:c %da como resultado una matriz 2x(mxn) a partir de la
           matriz xyz de 3x(mxn)
    vecxy(1,c1)=vectorxyz(1,c1)/coc(c1);
    vecxy(2,c1)=vectorxyz(2,c1)/coc(c1);
end

```

Como salida de esta función se obtiene una matriz de dimensiones 2 x 256 que contiene las componentes x y de cromaticidad para cada pixel de la imagen.

Una vez obtenidas las coordenadas se limita geoméricamente el área de verdes (correspondiente al color de las hojas de la planta) en el diagrama cromático a través de dos rectas y una circunferencia que tienen como origen el color blanco.

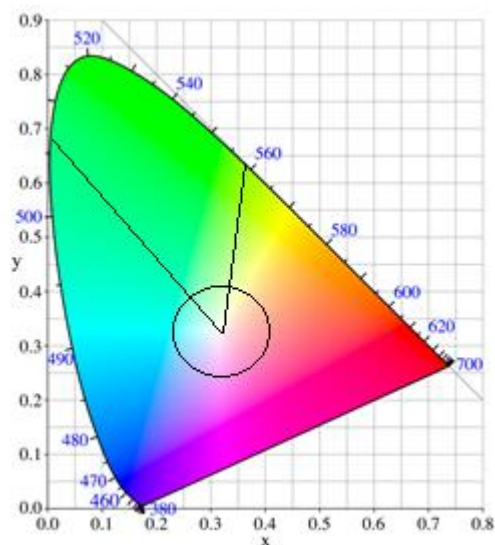


Figura 5.2: representación del filtro color en el diagrama cromático.

La sección de código Matlab donde se evalúa si las coordenadas de cada pixel, que se corresponden ahora con la nueva matriz de coordenadas, están o no incluidas en la zona de interés del diagrama cromático es la que se detalla a continuación:

```

a1=-1;
a2=1;
bx=0.315;
by=0.335;
r=0.01;
[f,c]=size(vecxy);
vcfiltra2=zeros(c,3); %predefino el tamaño de la matriz (mxn)x3 y
coloco en cero todos sus elementos

```

```

for i=1:c if(((vecxy(1,i)-bx)*a1<(vecxy(2,i)-by)) && ((vecxy(1,i)-
    bx)*a2<(vecxy(2,i)-by)) && ((vecxy(1,i)>sqrt(r^2-(vecxy(2,i)-
    by)^2)+bx))
vcfiltra2(i,:)=vc(i,:); %da como resultado el vector color filtrado
de (mxn)x3
end
end
mr1=vcfiltra2(:,1); %vector columna de (mxn)x1 con todas las
componentes del rojo
mg1=vcfiltra2(:,2);%vector columna de (mxn)x1 con todas las
componentes del verde
mb1=vcfiltra2(:,3);%vector columna de (mxn)x1 con todas las
componentes del azul
[m,n1]=size(map);n=n1/3;
mr=reshape(mr1,[m n]); %armo matriz rojo de mxn a partir del vector
rojo
mg=reshape(mg1,[m n]);%armo matriz rojo de mxn a partir del vector
verde
mb=reshape(mb1,[m n]);%armo matriz rojo de mxn a partir del vector
azul
%armo el mapa de mxnx3
mapfilttotal(:,:,1)=uint8(mr);
mapfilttotal(:,:,2)=uint8(mg);
mapfilttotal(:,:,3)=uint8(mb);
end

```

Para este caso, c es el valor máximo de la segunda coordenada de la matriz que en el caso de las imágenes empleadas como prueba fue 256. Los valores a_1 y a_2 son las pendientes de las rectas y bx y by son las coordenadas del color blanco en el diagrama cromático como se muestra en la figura 5.2. El valor “ r ” es el radio de la circunferencia. Si bien los valores teóricos de las coordenadas xy de cromaticidad para el blanco son $1/3$ y $1/3$, mediante pruebas con todas las imágenes utilizadas se llegó a los valores heurísticos $bx=0.315$ y $by=0.335$. De la misma manera, se obtuvieron los valores óptimos para las pendientes de las rectas y el radio de la circunferencia.

Resumiendo, si la coordenada del mapa de colores, que a su vez corresponde a un par xy en el diagrama cromático, se ubica dentro de la zona limitada por las rectas y circunferencia entonces el mapa de colores de la imagen filtrada mantiene los valores $sRGB$ originales; en caso contrario, en el mapa de colores filtrado se coloca negro, es

decir $R=0$, $G=0$ y $B=0$. Se realiza esta operación sobre el vector mapa de colores y no recorriendo cada pixel de la imagen, con el fin de optimizar el algoritmo en cuanto a velocidad y eficiencia.



a)



b)

Figura 5.3: ejemplo de filtrado en el diagrama cromático. a) Imagen de planta de soja original. b) Imagen resultante del filtrado color.

5.3.2. Binarización Otsu.

Una vez realizado el filtrado color, se procedió a la binarización de la imagen resultante, expresada en el modelo RGB . Para esta tarea se utilizó el canal G (verde), dado que éste contiene la información del objeto de interés. A este canal se lo trató como una imagen en niveles de gris.

Para el proceso de binarización primero se halla el umbral óptimo, utilizando el método adaptativo por ventana de Otsu, el cual fue descrito en el capítulo III (3.6.2), para luego obtener una imagen binarizada acorde a los requerimientos del algoritmo propuesto.

```
function [BW]= binarizacion_otsu(mapfilttotal)
Y=mapfilttotal(:,:,2);
Y=double(Y);
Y=Y/max(max(Y)); %cada elemento de la matriz Y la divido por el
elemento más grande de esa matriz (normaliza la imagen)
level=graythresh(Y); %halla el umbral de binarización
BW=im2bw(Y,level); %binariza la imagen
end
```

La imagen resultante de la binarización contiene al objeto de interés representado mediante un 1 (blanco), y al fondo representado con 0 (negro). Esta imagen contendrá cierto nivel de ruido tanto en el objeto de interés como en el fondo.



Figure 5.4: Imagen binarizada mediante el método adaptativo por ventana Otsu

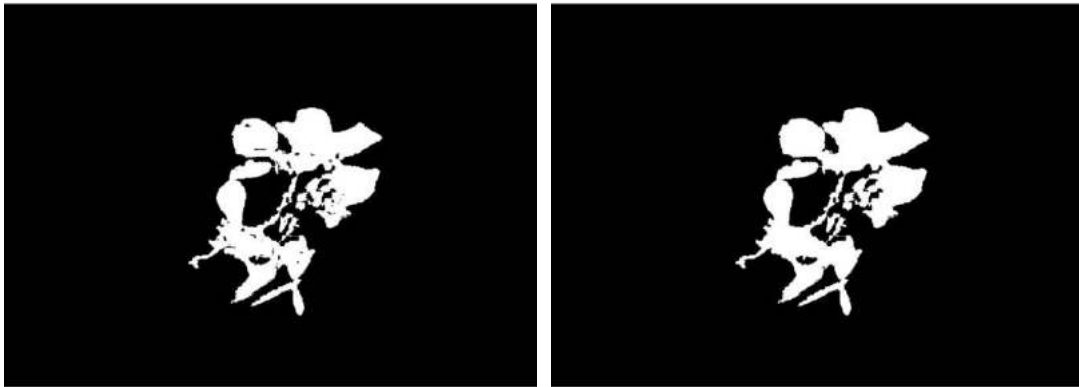
5.3.3. Eliminación de ruido.

Este paso consiste en la eliminación del ruido de los objetos de interés, por medio de la remoción de objetos de tamaño menor a 50 píxeles y el cierre de aquellos huecos en las hojas cuyo tamaño es inferior a 50 píxeles. Este valor fue seleccionado debido a que el tamaño mínimo de las ramificaciones y follaje de la planta, para todas las imágenes analizadas, fue superior a este valor. La identificación de objetos se realizó determinando la conectividad entre los píxeles de la imagen con un criterio de conectividad 8. En una imagen binaria, la conectividad 8 define una vecindad entre un píxel y los píxeles adyacentes a éste en las direcciones horizontal, vertical y las dos diagonales.

Una vez identificados los objetos contenidos en la imagen, se procedió al conteo de los píxeles presentes en cada uno de éstos, y luego se eliminaron aquellos con un tamaño inferior al indicado, de esta manera se obtuvo una nueva imagen binaria con el objeto de interés libre de ruido. Para esto se utilizó el operador *areaopen* definido en la sección de Morfología Matemática del Capítulo II

```
function [BW3]= eliminar_ruido(BW)
BW2 = bwareaopen(BW,50); %elimina de la imagen binarizada objetos que
contengan menos de 50 pixeles por fuera de las hojas
BW3=~BW2;
BW3=bwareaopen(BW3,50); % elimina de la imagen binarizada objetos
(huecos en las hojas) que contengan menos de 50 pixeles
```

```
BW3=~BW3;  
end
```



a)

b)

Figura 5.5: etapa de filtrado de ruido. a) eliminación de ruido externo a las hojas.
b) eliminación de ruido interno a las hojas.

5.3.4. Segmentación del follaje y cálculo del área foliar

Una vez eliminado el ruido interno y externo de las hojas, se procede a la identificación de los píxeles pertenecientes a cada una de las hojas y al conteo de los mismos. El número de píxeles de cada hoja corresponde al área foliar de la misma y puede ser transformado a unidades de área mediante la multiplicación por un factor de conversión que depende, entre otros factores, de la distancia entre la cámara y la planta.

Debido a que en algunos casos existe una superposición de las hojas, sobre todo en aquellas imágenes que contienen una vista superior de la planta, a las hojas superpuestas se las consideró como un único objeto.

```
function [leyenda x y BW3]=morfologia(BW2)  
  
imgsegmentada=bwlabeln(BW3); % enumera la cantidad de objetos  
p=max(max(imgsegmentada)); % devuelve la cantidad de objetos totales  
[m n]=size(imgsegmentada);  
suma=0;  
leyenda = {};  
x = [];  
y = [];  
for z=1:p  
contx=0;  
conty=0;
```



```

for i=1:m
for j=1:n
if imgsegmentada(i,j)==z
        contx=contx+i;
        conty=conty+j;
end
end
end
objeto(z)=sum(sum(imgsegmentada==z));
x(z)=round(contx/objeto(z));
y(z)=round(conty/objeto(z));
        leyenda{z}=num2str(objeto(z)); % convierte una matriz en una
cadena de numeros
        suma=suma+objeto(z);
end
        x(z+1)=m-25;
        y(z+1)=n-220;
        leyenda{z+1}=num2str(suma);
end

```

5.3.5. Bordes.

Finalmente para obtener el borde de los objetos de interés se utiliza el gradiente morfológico explicado en el capítulo III el cual consiste en restarle a la imagen original la imagen erosionada por un elemento estructurante de geometría y tamaño acorde a los objetos que conforman la imagen.

```

function imgout=bordes(BW3,map)

se3 = strel('disk',3); % crea un elemento estructurante (forma y
tamaño)
BW4 = BW3 - imerode(BW3,se3); % a la imagen se le resta la imagen
erosionada (quedan bordes de imagen resultado)
imgout=double(map); % imagen original
% imgout(:, :, 1) = max(imgout(:, :, 1), BW4*255); % no se incluye para
que se marque mejor el borde
imgout(:, :, 2) = max(imgout(:, :, 2), BW4*255); % forma una matriz
comparando elemento por elemento (de las 2 matrices) y toma el más
grande

```

```

imgout(:,:,3) = max(imgout(:,:,3),BW4*255); % queda bien marcado el
borde de la imagen resultado sobre la imagen original
imgout=uint8(imgout);
end

```

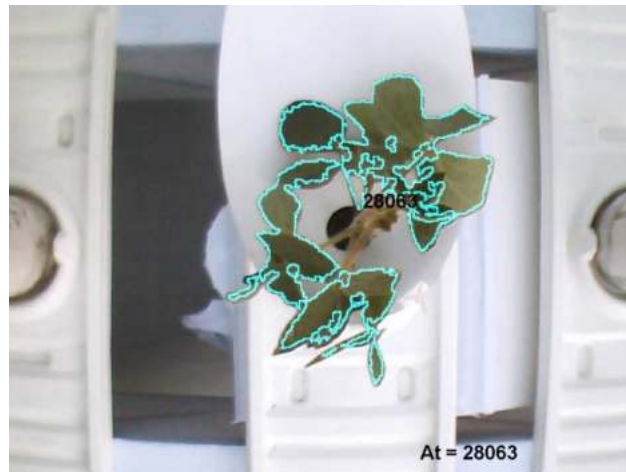


Figure 5.6: Imagen Segmentada

5.4. Programa final.

En esta sección se presenta el código de programa que incluye las funciones detalladas anteriormente, la carga de imágenes a procesar, el almacenamiento y presentación de las imágenes resultantes y el reporte de los resultados.

```

%%%%%%Procesamiento completo de un conjunto de imagenes.%%%%%%%%
clc;
clear all;
close all;
warning off;
%Borra del workspace las variables generadas anteriormente. Eliminar
todas las figuras generadas anteriormente.
%En el comando no aparecerán las advertencias que reclame MATLAB al
correr el algoritmo.

%ELEGIR DIRECTORIO PARA CARGAR IMAGENES.
try%se utiliza para probar si el comando siguiente genera error, si
se genera error se salta a catch.
[maindirname]=uigetdir(); %para abrir y elegir un directorio de
trabajo, si no elijo y selecciona "cancel" me devuelve valor = 0
catch

```

```

errordlg('NO SE HA SELECCIONADO UN DIRECTORIO','ERROR...'); %abre un
mensaje de error.
return
end
if isnumeric(maindirname) %devuelve true si es numérico
(maindirname).
return
end
contenido=dirsearch(maindirname);
imgextenread='.JPG';

%ELEGIR DIRECTORIO DONDE GUARDAR IMAGENES.
mainpath=uigetdir(); %para abrir y elegir un directorio de trabajo,
si no elijo y selecciona "cancel" me devuelve valor = 0
if isnumeric(mainpath) %devuelve true si es numerico (mainpath).
return
end
submainpath=namesearch(maindirname); %abre carpetas del "maindirname"
hasta tener solo archivos.
mkdir([mainpath '\\' submainpath]);

[pathname imgname]=imgsearch(contenido,maindirname,imgextenread);
if isempty(imgname)==0
    [m n]=size(imgname);
    for i=1:m
        %LEE LAS IMÁGENES A PROCESAR.

pathread=[maindirname '\\' cell2mat(pathname(i,1)) '\\'];
fileread=cell2mat(imgname(i,1));
map=imread([pathread fileread]);

%% PROCESAMIENTO RGB.

mapfilttotal = mapafiltradototal (map);
BW = binarizacion_otsu(mapfilttotal);
BW2 = eliminar_ruido(BW);
[leyenda x y BW3]=morfologia(BW2);
imgout = bordes (BW3,map);

%=====

```

```

%GUARDA LAS IMÁGENES PROCESADAS.

pathsave=[mainpath '\\' submainpath '\\' cell2mat(pathname(i,1)) '\\'];
filesave=cell2mat(imgname(i,1));
if isempty(dir([mainpath '\\' submainpath
    '\\'cell2mat(pathname(i,1))]))==1
    mkdir([mainpath '\\' submainpath],cell2mat(pathname(i,1)));
end
% ----
figure
set(gcf,'Name','IMAGEN ORIGINAL');
imshow(map);
set(gca,'position',[0 0 1 1]);
truesize;
    print('-djpeg90',[pathsave
strrep(lower(filesave),'.jpg','_A_original.jpg')]);
    figure
    set(gcf,'Name','IMAGEN FILTRADA EN RGB');
    imshow(mapfilttotal);
    set(gca,'position',[0 0 1 1]);
    truesize;
    print('-djpeg90',[pathsave
strrep(lower(filesave),'.jpg','_B_filteredRGB.jpg')]);
    figure
    set(gcf,'Name','IMAGEN BINARIZADA');
    imshow(BW);
    set(gca,'position',[0 0 1 1]);
    truesize;
    print('-djpeg90',[pathsave
strrep(lower(filesave),'.jpg','_C_binary.jpg')]);
figure
    set(gcf,'Name','ELIMINACIÓN DE RUIDO');
imshow(BW2);
    set(gca,'position',[0 0 1 1]);
    truesize;
    print('-djpeg90',[pathsave
strrep(lower(filesave),'.jpg','_D_filteredNoise.jpg')]);
    figure
    set(gcf,'Name','SEGMENTACION');
    imshow(BW3);
    set(gca,'position',[0 0 1 1]);

```

```

        truesize;
        print('-djpeg90',[pathsave
strrep(lower(filesave),'.jpg','_E_Segmentacion.jpg')]);
        figure
        set(gcf,'Name','IMAGEN RESULTANTE');
        imshow(imgout);
        t=length(leyenda);
        text(y(1:t-1),x(1:t-1),leyenda(1:t-
1),'FontSize',18,'Color',[0 0 0],...
'FontWeight','bold');
        text(y(t),x(t),['At = '
cell2mat(leyenda(t))'],'FontSize',18,'Color',[0 0 0],...
'HorizontalAlignment','left','FontWeight','bold');
        posf=get(gcf,'Position');
        posa=get(gca,'Position');
        set(gcf,'position',[posf(1,1) posf(1,2) posf(1,3)*posa(1,3)
posf(1,4)*posa(1,4)]);
        set(gca,'position',[0 0 1 1]);
        truesize;
        print('-djpeg90',[pathsave
strrep(lower(filesave),'.jpg','_F_processed.jpg')]);
% ----
        close all;

end
end

```

En el siguiente capítulo se presentarán las pruebas de validación del algoritmo, que consistieron en aplicar el mismo a la medición del follaje de aproximadamente cincuenta plantas de varios genotipos de soja, en diferentes etapas de crecimiento y bajo distintas condiciones de adquisición, es decir alta variación en los niveles de brillo y sombras.

CAPÍTULO VI

RESULTADOS

Capítulo VI

RESULTADOS

Para cuantificar la eficiencia de un método de segmentación, es necesario llevar a cabo experimentos de validación, lo cual consiste en comparar el resultado obtenido con algún modelo real [27][28][29]. La manera más directa para comparar una segmentación es realizar una simple inspección visual entre la segmentación realizada y una obtenida manualmente por un especialista. Sin embargo, esto no garantiza una validación debido a la gran variabilidad inter e intra-usuario.

Una vez obtenida la segmentación de referencia, es necesario establecer alguna medida de similitud para cuantificar la precisión y exactitud del método de segmentación.

Existe en la bibliografía un gran número de medidas o índices de similitud, entre los más utilizados se pueden mencionar:

Coefficiente de Tanimoto (TC , *Tanimoto Coefficient*), se define como:

$$TC(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (6.1)$$

donde $| \cdot |$ denota el cardinal del conjunto [28][30][31][32][33]. La medida de Tanimoto es la razón entre el número de elementos que los vectores tienen en común y el número de elementos distintos. En el caso particular de variables binarias, las entradas de los dos vectores se pueden resumir en la siguiente tabla:

	A	0	1
B	0	a	b
	1	c	d
Figura 6.1.: Entradas de dos vectores			

donde a representa el número de posiciones donde los vectores A y B coinciden en tomar el valor 0. Similarmente b representa el número de coincidencias donde A y B valen ambos 1. Mientras que c y d representan el número de no coincidencias. En función de esto la medida de Tanimoto se reduce a:

$$TC(A, B) = \frac{a + d}{a + 2(c + b) + d} \quad (6.2)$$

Coefficiente de Dice. A partir del coeficiente de Tanimoto se define el coeficiente de Dice de la siguiente manera:

$$DICE(A, B) = \frac{2TC(A, B)}{1 + TC(A, B)} \quad (6.3)$$

Coefficiente de exactitud. La medición de exactitud (ACC , *Accuracy*) en la segmentación tiene en cuenta la cantidad de píxeles que han sido correctamente clasificados con respecto a la cantidad total de píxeles que se esperaba detectar. Este coeficiente se define como:

$$ACC(A, B) = \frac{|A \cap B|}{B} \quad (6.4)$$

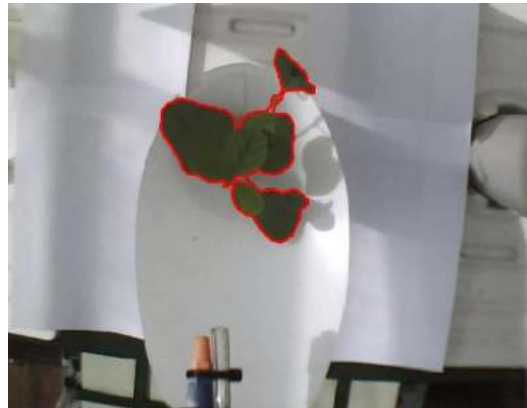
Para esta aplicación se optó por los índices de **Tanimoto (6.1)** y de **Exactitud (6.4)**, ya que la clasificación involucra sólo dos clases: figura y fondo.

A continuación, se ilustrarán los resultados finales al aplicar el método propuesto en esta tesis. En la siguiente figura, pueden observarse los resultados del procesamiento de seis plantas elegidas arbitrariamente. En la primera columna se muestra la imagen original, en la segunda la segmentación objetivo o “Gold Standard” (segmentación de referencia), en la tercer y cuarta columna el resultado de aplicar el filtro color en el modelo RGB y los obtenidos al emplear un filtro HSI (con sus respectivos coeficientes de Tanimoto y de exactitud). Adicionalmente se realizó una superposición de los bordes de las hojas segmentadas a la imagen original incluyendo el valor del área en píxeles en el centro de cada hoja. También, en la parte inferior derecha de cada imagen se muestra la suma de todas las áreas, correspondiente al área foliar de la planta para la vista correspondiente.

Imagen original



Gold Standard

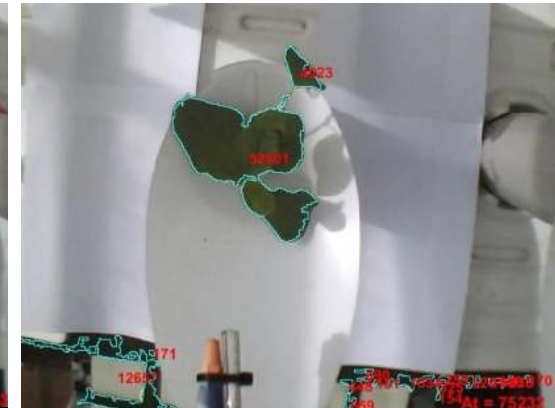


Método RGB

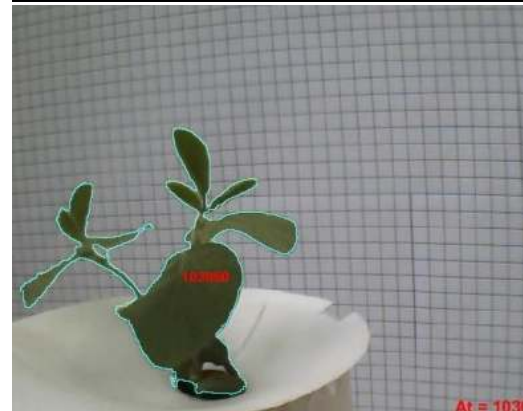
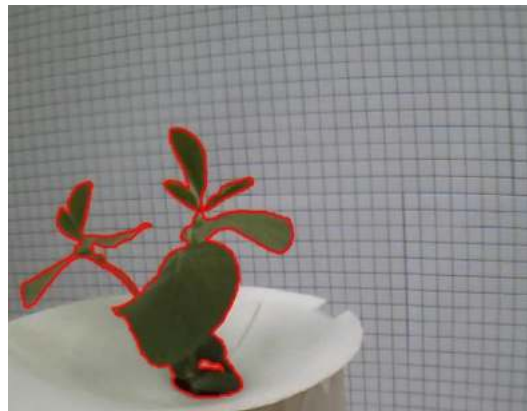


Tanimoto: 0,931 Exactitud: 0,957

Método HSI



Tanimoto: 0,655 Exactitud: 0,936



Tanimoto: 0,901 Exactitud: 0,942



Tanimoto: 0,890 Exactitud: 0,895

Imagen original



Gold Standard

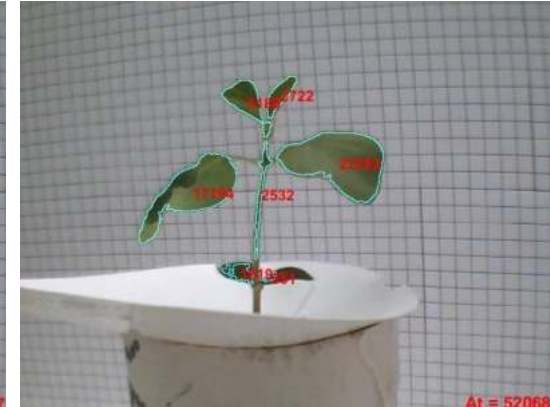


Método RGB



Tanimoto:0,894 Exactitud: 0,913

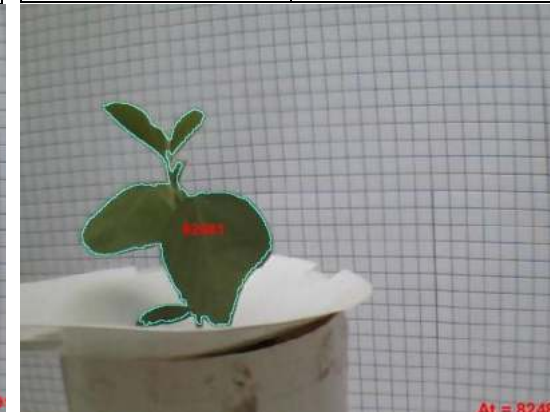
Método HSI



Tanimoto:0,839 Exactitud:0,841



Tanimoto:0,950 Exactitud:0,971



Tanimoto:0,930 Exactitud:0,935

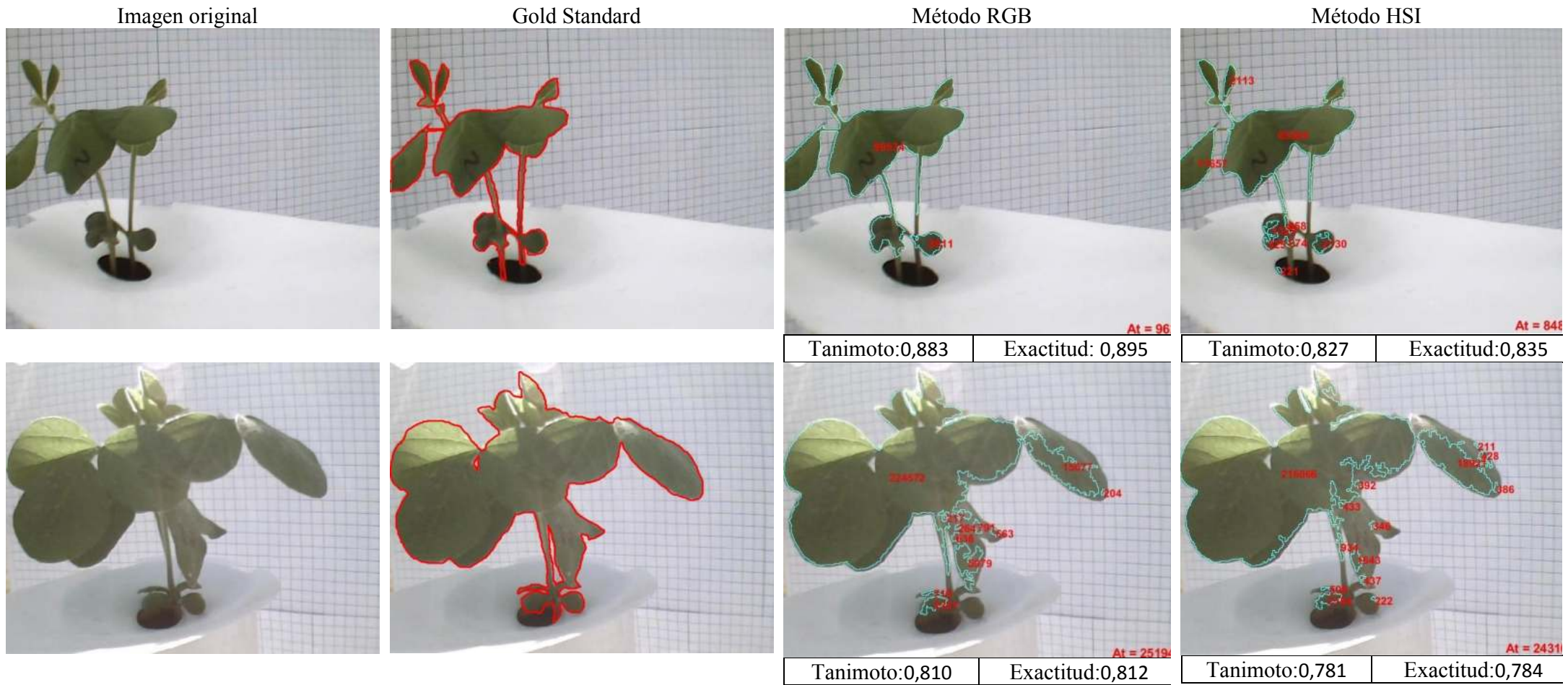


Figura 1: Resultados obtenidos en seis imágenes elegidas aleatoriamente.

Los valores obtenidos para los índices calculados son altos, aun en las peores condiciones de iluminación, lo que permite asegurar la robustez del método propuesto.

CAPÍTULO VII

CONCLUSIONES

Capítulo VII

CONCLUSIONES

Al comienzo de este proyecto se estableció como objetivo la realización de un algoritmo de procesamiento digital de imágenes en color que permitiera segmentar estructuras para su posterior cuantificación. Analizando los resultados se puede decir que:

- Se logró desarrollar, a través de Matlab, un algoritmo que permite determinar, de manera automática, el área foliar de una planta de soja. Dicho algoritmo, basado en un filtrado color en el diagrama cromático, puede discriminar objetos de color verde del fondo de la imagen.
- Se logró definir un apropiado protocolo para la captura de imágenes, permitiendo procesar grandes grupos de imágenes, con un bajo tiempo de procesamiento, lo cual, junto con la adecuada presentación de los resultados, facilita el análisis y comparación del crecimiento de diferentes genotipos de soja.
- Manteniendo una distancia fija en la captura de las imágenes, se consiguió calcular el área foliar en base al número de píxeles que componen las hojas segmentadas. Los resultados obtenidos fueron validados utilizando los coeficientes de similitud.
- Dada la relativa simplicidad de las operaciones involucradas, el método propuesto puede ser programado en cualquier lenguaje sin necesidad de motores de cálculo ni cualquier otra librería específica. Esto abre la posibilidad de implementarlo en hardware, por ejemplo, a través de software insertado en un microprocesador.

ANEXO (PROYECCION A FUTURO)

Este proyecto de tesis comenzó como parte de un trabajo en conjunto con la Unidad Integrada Balcarce. Por razones ajenas al desarrollo de esta tesis, el convenio en el cual se enmarcaba no se concluyó. Por esta razón no se logró cumplir con la totalidad de los objetivos inicialmente planteados para este convenio, sin afectar de manera significativa el poder cumplir con los objetivos planteados en esta tesis como parte del proyecto macro. Principalmente no se pudo desarrollar el hardware para la utilización del algoritmo en tiempo real propuesto en esta tesis, fundamentalmente por falta de financiamiento y la excesiva extensión en los plazos previstos originalmente. Este hecho, tuvo un impacto negativo en cuanto a la posibilidad de contar en forma ininterrumpida con imágenes de las plantas en diferentes etapas de crecimiento y bajo distintas condiciones de adquisición, lo cual permitiría optimizar los valores de los parámetros que intervienen en el algoritmo propuesto en esta tesis.

A futuro, considerando lo anteriormente expuesto, y dando continuidad a la tarea desarrollada hasta esta instancia y teniendo en cuenta la importancia de la aplicación, queda pendiente la creación de un sistema robótico que capture las imágenes de las plantas para su procesamiento con el algoritmo propuesto en esta tesis. A modo de ejemplo se propone el siguiente anteproyecto para el desarrollo de la plataforma en la cual se incorporaría el algoritmo

Plataforma robot

La plataforma robot se piensa con una o más cámaras incorporadas, se desplazaría por un monorraíl capturando las imágenes de las plantas en forma individual, siendo la identificación de cada planta mediante un código QR. En la siguiente figura queda ilustrado el proceso mencionado:

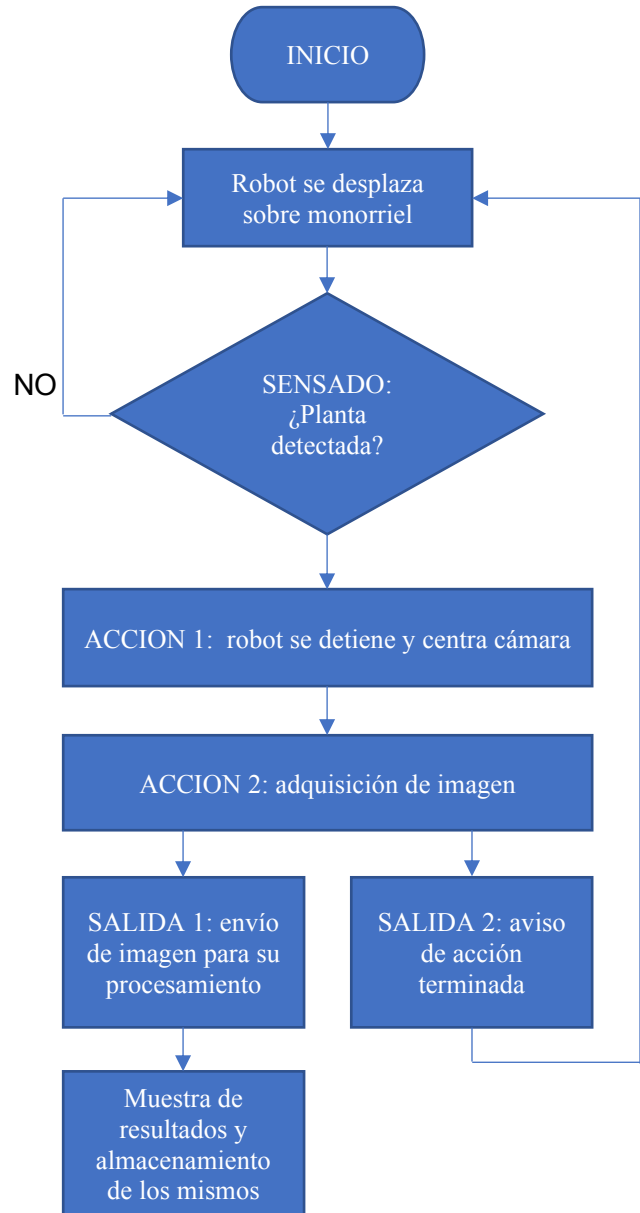
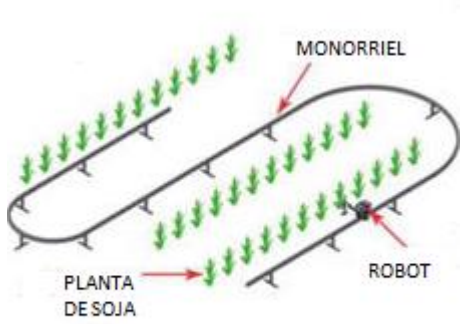


Figura 7.1. Plataforma robot.

CAPÍTULO VIII

REFERENCIAS

Capítulo VIII

REFERENCIAS

- [1] González, R., Woods, R., y Eddins, S.: "Digital Image Processing", 2002.
- [2] Serra, J. "Image Analysis and Mathematical Morphology", 1982.
- [3] Ronse C. and Heijmas H., (1991). "The algebraic basis of mathematical morphology. Part II: Openings and closings", *Comp. Vision Graph. Image Process: Image Understanding*, 54, 74-97.
- [4] Facon J., (1996). "Morfología Matemática. Teoría y ejemplos", Curitiba Brasil, CITS.
- [5] Bangham, J.A. and Marshall S., (1998). "Image and Signal processing with mathematical morphology," *IEE Electronics & Communication Engineering Journal*, 10, 117-128.
- [6] Dougherty, E. (1992): "An Introduction to Morphological Image Processing", (SPIE, Washington).
- [7] Dougherty E. R., (1993). "Mathematical Morphology in Image Processing", CRC Press. ISBN 0-8247-8724-2.
- [8] Braga-Neto U., (2004). "Multiscale Connected Operators", Submitted to JM IV Special Issue on Mathematical Morphology after 40 Years, 2004.
- [9] Vincent, L.: "Morphological Grayscale Reconstruction in Image Analysis: Applications and efficient Algorithms" , 1993.
- [10] Serra, J. and Vincent, L.: "An overview of morphological filtering," *IEEE Trans. Circuits, Systems and Signal Proc.*, 1992.
- [11] Heijmans H., (1997). "Connected Morphological Operators for Binary Images", *Probability Networks and Algorithms (PNA)*, PNA-R9708.
- [12] Pham D. L., Xu C., Prince J. L., (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, Vol. 2, pp. 315-37. doi: 2/1/315 [pii] 10.1146/annurev.bioeng.2.1.315..
- [13] Geman, S. and Geman, D., (1984). "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. PAMI-6, pp. 721-741.

[14] Skarbek y Koschan 1994.

[15] González, R. y Woods, R., (1996). "Tratamiento Digital de imágenes", Addison Wesley.

[16] Jianping, D., Yau, K., Elmagarmid, Ahmed and Aref, Walid. (2001). Automatic image segmentation by integrating color-edge extraction and seeded region growing. En: IEEE TransactionsonImage Processing. Vol. 10, no. 10; p. 1454–1465.

[17] Betancur, J. (2002). Segmentación de granos de café mediante métodos de crecimiento de regiones. Manizales, 89 h. Tesis Ingeniería Electrónica. Universidad Nacional de Colombia. Facultad de Ingeniería y Arquitectura.

[18] Sahoo P. K., Soltani S., Wong A. K. C., Chen Y. C., (1988). "A survey of thresholding techniques". Computer Vision, Graphics, and Image Processing, Vol. 41 (2), pp. 233-260. doi: 10.1016/0734-189X(88)90022-9.

[19] Zijdenbos A., Dawant B. M., Margolin R. A., and Palmer A. C., (1994). "Morphometric Analysis of White Matter Lesions in MR Images: Method and Validation", IEEE Transactions on Medical Imaging, 13(4), pp. 716-724.

[20] Jain, A., (1998). "Fundamentals of Digital Image Processing", Prentice Hall, Englewood Cliffs, New Jersey.

[21] Saghri, J.A., and Freeman, H., (1981). "Analysis of the Precision of Generalized Chain Codes for the Representation of Planar Curves, IEEE Journal of Pattern Recognition and Machine Intelligence, No. 5, pp. 533-539.

[22] Fan J., Zeng G., Body M., Hacid M.-S. (2005). "Seeded region growing: an extensive and comparative study". Pattern Recognition Letters, Vol. 26 (8), pp. 1139-1156.

[23] Castellanos R., Mitra S., (2000). "Segmentation of Magnetic Resonance Images Using a Neuro-Fuzzy Algorithm". En proceedings de 13th IEEE Symposium on Computer-Based Medical Systems (CBMS '00). pp. 207. Washington, DC, USA, IEEE Computer Society. ISBN 0-7695-0484-1.

[24] SCHANDA, J. (2007). Colorimetry: Understanding the CIE System. Wiley, New York.

[25] Salomon, D.: "Data compression: The complete reference and Appendixes", 2000.

[26] F. G. Tomasel , J. M. Paruelo , G. Abras , V. Ballarin , and E. Moller (2001)“ A chromaticity -based technique for estimation of above-ground plant biomass ” – F. G. Tomasel , J. M. Paruelo , G. Abras , V. Ballarin , and E. Moller – Journal of Applied Vegetation Science Vol 4: 207-212, ISSN 1402

[27] Barra V., Lundervold A., (2007). “A Collaborative Software Tool for the Evaluation of MRI Brain Segmentation Methods”. En proceedings de Lundervold, A. (Ed.) International Symposium on Information Technology Convergence (ISITC 2007). pp. 235-239. ISBN 978-0-7695-3045-1.

[28] Bouix S., Martin-Fernandez M., Ungar L., Nakamura M., Koo M. S., Mccarley R. W., Shenton M. E. (2007). “On evaluating brain tissue classifiers without a ground truth”. Neuroimage, Vol. 36 (4), pp. 1207-1224. doi:10.1016/j.neuroimage.2007.04.031.

[29] Crum W. R., Camara O., Hill D. L. G., (2006). “Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis”. IEEE Transactions on Medical Imaging, Vol. 25 (11), pp. 1451-1461.

[30] Jimenez-Alaniz J. R., Medina-Banuelos V., Yanez-Suarez O. (2006). “Data driven brain MRI segmentation supported on edge confidence and a priori tissue information”. IEEE Transactions on Medical Imaging, Vol. 25 (1), pp. 74-83.

[31] Santalla H., Meschino G. J. and Ballarin V. L., (2007). “Effects on MR images compression in tissue classification quality”. Journal of Physics: Conference Series, Vol. 90. doi: 10.1088/1742-6596/90/1/012061.

[32] Song T., Angelini E. D., Mensh B. D., Laine A., (2004). “Comparison study of clinical 3D MRI brain segmentation evaluation”. En proceedings de Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 3, pp. 1671-1674. San Francisco, CA, USA.

[33] Song T., Jamshidi M. M., Lee R. R., Huang M., (2007). “A Modified Probabilistic Neural Network for Partial Volume Segmentation in Brain MR Image”. IEEE Transactions on Neural Networks, Vol. 18 (5), pp. 1424-1432.