DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE VERIFICACIÓN DE DESPACHO DE MEDICAMENTOS POR MEDIO DE VISIÓN POR COMPUTADOR

Diego Armando Ariza Ortiz

Selim Alberto Saba Santiago

ESCUELA DE INGENIERIA Y ADMINISTRACION
FACULTAD DE INGENIERIA ELECTRONICA
Bucaramanga
2013

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE VERIFICACIÓN DE DESPACHO DE MEDICAMENTOS POR MEDIO DE VISIÓN POR COMPUTADOR

Proyecto de Grado

Diego Armando Ariza Ortiz

Selim Alberto Saba Santiago

DIRECTOR DEL PROYECTO
CESAR A. ACEROS MORENO

ESCUELA DE INGENIERIA Y ADMINISTRACION
FACULTAD DE INGENIERIA ELECTRONICA
Bucaramanga
2013

Nota de Aceptación

_____

_____

_____

_____

Firma del Presidente del Jurado

_____

Firma del Jurado

_____

Firma del Jurado

_____

A Dios, por guiarme en todo momento de mi vida y por regalarme una bendición cada día. A mi familia especialmente a mis padres que siempre han estado a mi lado y finalmente a mis amigos con quienes he compartido una excelente etapa de mi vida.

Diego Armando Ariza Ortiz

A mis padres, por toda la confianza, el cariño y apoyo durante toda mi vida. A mis hermanos por brindarme momentos de alegría.

Selim Alberto Saba Santiago

AGRADECIMIENTOS

Los autores quisieran agradecer al MSc. Cesar Augusto Aceros Moreno por su dirección y colaboración a lo largo del desarrollo del proyecto, a Offimedicas S.A. por su apoyo, a la Universidad Pontificia Bolivariana, y especialmenete Dios y a nuestras familias por darnos la oportunidad de cumplir esta meta.

TABLA DE CONTENIDO

pagina

LISTA DE TABLAS

Tabla pagina

LISTA DE FIGURAS

Figura pagina

GLOSARIO

Picking: es el proceso logstico que consiste en la seleccin y recoleccin de materiales ubicados en una unidad de almacenamiento superior la cual contiene ms unidades que las extradas. Este proceso es bsico en la preparacin de pedidos ya que puede afectar directamente la productividad.

Simbologa: en los cdigos de barras, es la informacin almacenada por las barras y los espacios con distintos ordenamientos.

FLAG: es el conjunto de 2 o 3 dgitos del cdigo de barras que se usan para la identificacin internacional del pas de origen de cada producto.

EAN 13: es un sistema de codificacin constituido por barras y espacios paralelos de ancho variable, el nmero de barras y espacios en donde se codifica la informacin es fijo.

EAN 8: es la versin reducida del sistema EAN 13.

UPC A: (Universal Product Code) tipo de cdigo popular en la industria norteamericana que utiliza barras y espacios para codificar informacin de 12 caracteres.

UPC E: (cdigo reducido) versin de cdigo UPC llamado cero suprimido ya que elimina 4 ceros de la simbologa.

Diafragma: Disco pequeo horadado, situado en el objetivo de la cmara, que sirve para regular la cantidad de luz que se ha de dejar pasar. RAE

Fotosensible: material que reacciona al ser expuesto a la luz.

Infrarrojo: se dice de la radiacin del espectro electromagntico de mayor longitud de onda que el rojo y de alto poder calorfico. La longitud de onda de esta radiacin

esta entre 0,7 y 1000 micrmetros, y es emitida por cualquier cuerpo que tenga una temperatura mayor a 0 Kelvin.

RGB: es la composicin del color en trminos de la intensidad de los colores primarios rojo, verde y azul.

Saturacin: es la intensidad de color en una foto, entre mayor sea el grado de blanco, gris o negro en un color, menor ser el valor de la saturacin.

Sistema operativo: es el software que se encarga de gestionar el uso del hardware entre diferentes programas de aplicacin del computador. http://www.euram.com.ni

Linux: es un ncleo libre de sistema operativo basado en Unix.

Unix: es un sistema operativo multiusuario y multitarea.

Arquitecturas: estructura lgica y fsica de los componentes de un computador.

GTK+: (GIMP Toolkit) es un conjunto de herramientas multiplataforma para la creacin de interfaz graficas de usuario.

Driver: es el software que le permite al sistema operativo interactuar con un perifrico, en otras palabras se conoce como controlador.

UVC: es un controlador de tipo USB especializado en los perifricos de entrada de imagen. revisar

Luvcview: herramienta que permite visualizar las webcams basadas en USB el cual incluye un decodificador mjpeg y es capaz de almacenar video en formato AVI. http://www.marlonj.com

Multiplataforma: Dicho de una aplicacin o de un producto informtico: Que puede ser utilizado por distintos sistemas o entornos.

GUI: (interfaz grfica de usuario) es un software que le permite al usuario interactuar con el ordenador.

C++: es un lenguaje de programacin basado en el lenguaje C, se caracteriza por ser verstil, potente y popular entre los programadores profesionales.

Relacional: en bases de datos, relacional se refiere a las bases que cumplen con el modelo relacional, el cual se trata de modelo lgico que establece una estructura de datos de tal forma que la informacin del mundo real se representa de manera intuitiva. Este modelo se aplica en la mayora de sistemas de gestin de datos por su versatilidad y potencia.

ABSTRACT

TITLE: DESIGN AND CONSTRUCTION OF A DRUG DISPATCH VERIFICATION SYSTEM WITH COMPUTER VISION

BY: DIEGO ARMANDO ARIZA ORTIZ
SELIM ALBERTO SABA SANTIAGO

DEPARTAMENT: ELECTRONIC ENGINEERING DEPARTMENT

CHAIR: CESAR A. ACEROS MORENO

KEYWORDS: Computer Vision, Barcode, OpenCV, ZBAR, Business Logistics

Abstract

The project consists in the design and the construction of system for the verification of medication dispatches based in computer vision. The system is composed of a conveyor belt, a digital camera, artificial light and a computer as data processor. The medications are displaced on the conveyor belt, located in such a way that the barcodes are always facing the same way, in order that the camera may capture images that contain this information. The images captured and stored will be processed in the computer in order to compare between the billed products and those that are about to the dispatched. The process stars with an image that of approximately two megapixels which contains the barcode, the first stage of the program seeks to select exclusively the area where the code is found, this with the purpose of reducing the size of the image to a minimum. In this stage are used tools from the open source Open CV library to apply functions such as border detection and morphological operations, which are necessary to locate the bars and spaces of the image. In the second stage, cropped images are sent to a software specialized in decoding information with different formats, specially the most common systems for barcode codification. This program is open source and it is named ZBar. ZBars

answer is sent again to central processing, where a list is created with each one of the symbologies read and finally tis list is compared with the one solicited by the customer. In the event of any different and alert is generated.

RESUMEN

TITULO: DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE VERIFICACIÓN DE DESPACHO DE MEDICAMENTOS POR MEDIO DE VISIÓN POR COMPUTADOR

POR: DIEGO ARMANDO ARIZA ORTIZ
SELIM ALBERTO SABA SANTIAGO

DEPARTAMENTO: FACULTAD DE INGENIERÍA ELECTRÓNICA

DIRECTOR: CESAR A. ACEROS MORENO

PALABRAS CLAVE: **Visión por Computador, Código de Barras, OpenCV, ZBAR, Logística Empresarial**

El proyecto consiste en el diseño y la construcción de un sistema de verificación de despachos de medicamentos basado en la visión por computadora. El sistema está compuesto por una banda transportadora, una cámara digital, iluminación artificial y un computador como procesador de datos. Los medicamentos se desplazarán sobre la banda transportadora, ubicados de tal forma que los códigos de barras siempre estén en la misma orientación, con el fin de que la cámara pueda captar imágenes que contengan esta información. Las imágenes capturadas y almacenadas se procesarán en el computador con el fin de hacer la comparación entre los productos facturados y los que se van a despachar. El procesamiento inicia con una imagen de aproximadamente dos megapíxeles que contiene el código de barras, la primera etapa del programa busca seleccionar exclusivamente el área donde se encuentra el código, esto con el fin de reducir al mínimo el tamaño de la imagen. En esta fase se utilizan herramientas de la librería de código abierto OpenCV para aplicar funciones como detección de bordes y operaciones morfológicas, las cuales son necesarias para lograr ubicar las barras y espacios en la imagen. En la segunda etapa, se envían las imágenes recortadas a un software especializado en decodificar información con diferentes formatos, específicamente los sistemas más comunes de codificación de

códigos de barras. Este programa es de código abierto y se llama ZBar. La respuesta de ZBar se envía de nuevo al procesamiento central, en donde se crea una lista con cada una de las simbologías leídas y finalmente se compara este listado con el solicitado por el cliente. En caso de existir alguna diferencia, se genera una señal de alerta.

CAPITULO 1
INTRODUCCIÓN

La logística empresarial es el diseño de procesos y la realización de las operaciones para satisfacer la necesidad de un producto de forma adecuada. Los procesos claves son el servicio al cliente, el procesamiento de los pedidos, el transporte y la gestión de inventarios [1].

En el despacho de pedidos de la bodega de la empresa Offimedicas S.A. se manejan alrededor de tres mil tipos de medicamentos en grandes volúmenes. Al realizar los envíos se procura que el cliente esté lo más satisfecho posible, para ello se lleva a cabo una verificación manual a lo largo de todo el procesamiento de los pedidos. Este control se fundamenta en revisiones de cada uno de los pasos que se realizan desde que llega la solicitud del cliente hasta que se entrega el pedido. A pesar de que diariamente el número de clientes no es tan elevado, generalmente los pedidos abarcan diferentes ítems en grandes cantidades. Para la revisión de estos despachos un trabajador tiene que verificar una a una las cantidades de los productos, lo cual implica una inversión importante de tiempo y al final se pueden presentar errores que afectan tanto al cliente como a la empresa.

Estos errores afectan desde diferentes puntos el desempeño de la empresa, ya que pone en riesgo su credibilidad con los clientes, generan pérdidas económicas y trastorna todo el control del inventario de la bodega.

Para solucionar estos problemas, existen opciones en la industria[1] que no encajan en el perfil de Offimedicas S.A. El proyecto que aquí se plantea pretende corregir esta situación de forma innovadora y a un bajo costo, ya que se busca emplear un sistema de visión por computadora que haga la labor del conteo para la revisión de las cantidades.

El sistema obtendrá imágenes de los códigos de barras de cada uno de los productos a verificar, por medio de una cámara fotográfica digital, que enviará las imágenes a un computador dedicado para el procesamiento de las mismas. Dicho procesamiento consistirá en la lectura del código y su verificación comparándolo con los códigos requeridos por la red de Offimedicas S.A. El sistema indicará los errores, si los hay, de forma visual. El resultado de la evaluación será enviado a la red de la empresa. Ver Figura 1–1.

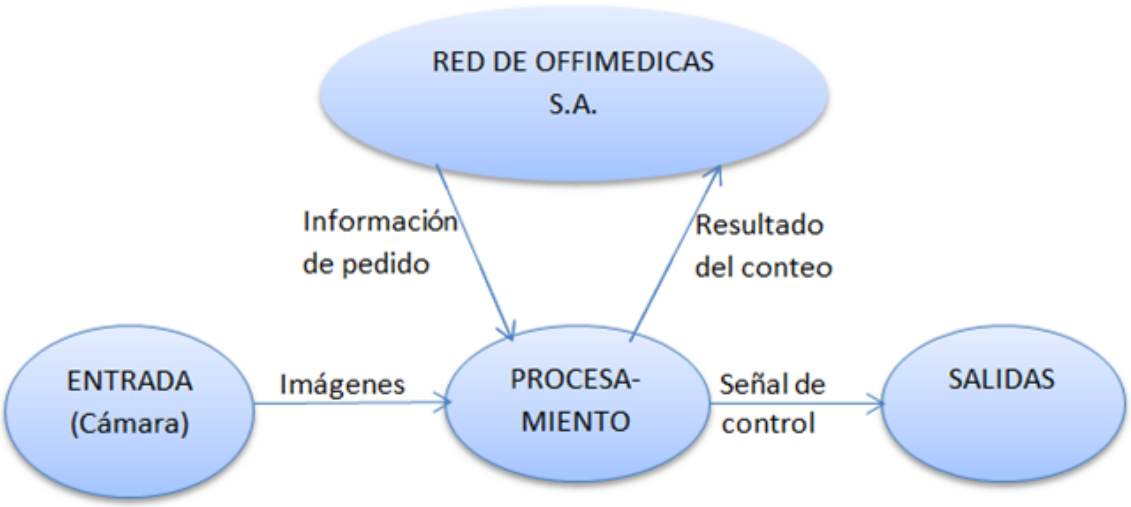

Figure 1–1: Diagrama de descripción del proceso

[1] KNAAP AG soluciones para logística y almacenaje http://www.knapp.com/cms/cms.php?pageName=glossary&iD=62

CAPITULO 2
PLANTEAMIENTO DEL PROBLEMA.

2.1 DEFINICIÓN DEL PROBLEMA

El proceso de despacho de grandes volúmenes de medicamentos de la empresa OFFIMEDICAS S.A. consta de tres etapas fundamentales: el picking [2], la verificación y la entrega. En el picking un operario ingresa el código de despacho en una terminal remota, la cual le indica la ubicación del medicamento en la bodega. Este debe identificarlo mediante el código de barras, extraer la cantidad requerida y continuar con el proceso hasta completar la totalidad de los artículos del pedido. Luego los productos seleccionados son entregados al área de verificación. En esta etapa otro trabajador recibe los medicamentos previamente seleccionados. Éste desconoce la cantidad de la orden de despacho, procede a contarlos e ingresa el resultado del conteo en el sistema. El dato ingresado por el operario se compara con el dato del pedido y en caso de existir diferencia se informa la irregularidad, sin revelar la cantidad de la orden, para ser corregida. En la tercera fase se hace entrega del pedido a una empresa transportadora que se encarga de llevarlo al cliente. Este proyecto interviene en la segunda etapa del proceso de despacho, la verificación, en donde se pretende reducir el tiempo de esta actividad y aumentar la confiabilidad en el resultado. El sistema planteado consiste en el conteo de productos farmacéuticos mediante visión por computador y una banda transportadora.

2.2 OBJETIVOS

2.2.1 General

Disear y construir un sistema que permita la verificación de cantidades de productos en una orden de despacho usando técnicas de procesamiento de imágenes.

2.2.2 Específicos

- Seleccionar los equipos adecuados para la aplicación propuesta.
- Disear un plano del sistema de conteo de productos.
- Disear la comunicación entre el sistema propuesto y el sistema de información existente en la empresa.
- Realizar pruebas del funcionamiento y desempeo de los dispositivos seleccionados, para verificar si son adecuados para esta aplicación.
- Construir el prototipo del equipo.

2.3 JUSTIFICACIÓN

Solucionar el problema que se presenta al enviar al cliente una cantidad de ítems incorrecta, o incluso medicamentos diferentes a los facturados. Ya que si el despacho contiene menor cantidad de productos, es necesario reenviar el faltante y en el caso de que contenga mayor cantidad, podría implicar pérdida de los productos sobrantes. En ambas situaciones se genera dificultad de mantenimiento del inventario, lo cual trastorna el sistema de abastecimiento por fallos en el registro de existencia. Esto genera sobrecostos, incumplimiento a los clientes debido a ventas superiores a la existencia, deterioro de la confiabilidad e imagen de la empresa y afecta el indicador de oportunidad.

CAPITULO 3
MARCO TEÓRICO

3.1 VISIÓN POR COMPUTADOR

La visión por computador incluye métodos para adquirir, procesar, analizar y entender imágenes y, en general, datos del mundo real, con el propósito de producir información en forma de decisiones o nuevas representaciones. El procesamiento consta de la manipulación de las imágenes, en forma de señales digitales, para obtener información implícita en la imagen. El análisis se encarga de obtener estructuras como bordes o regiones y determinar la relación entre estas estructuras. En un sistema de visión, el computador recibe una matriz de números de una cámara o una imagen en una memoria. Es trabajo del computador darle una interpretación a estos números con el fin de lograr una percepción de la escena. Hay varias herramientas mediante las cuales se puede hacer el análisis para la visión por computador, entre estas herramientas se encuentra la librería OpenCV que se usa en lenguajes de programación como C, C++ y Python.

3.2 OPENCV

Es una librería de código abierto que contiene más de 500 algoritmos optimizados para el análisis de videos e imágenes. Desde sus inicios en 1999 ha sido mundialmente adoptado como la principal herramienta de desarrollo por la comunidad de investigadores y desarrolladores en visión por computador. OpenCV fue desarrollado originalmente en Intel por el equipo que dirigía Gary Bradski, como una iniciativa para avanzar en la investigación de la visión por computador y promover el desarrollo

de aplicaciones basadas en esta. Esta herramienta fue diseñada para la eficiencia computacional y con fuerte enfoque en aplicaciones en tiempo real. Está escrita y optimizada para el lenguaje C y puede tomar ventajas de procesadores multinúcleo. Sus funciones abarcan áreas de trabajo como robótica y calibración de cámaras, interface de usuario, visión estéreo, seguridad, medicina e inspección de productos en fábricas[3].

Para el desarrollo de este proyecto se usaron las siguientes funciones provistas por OpenCV

3.2.1 Suavizado

Suavizado, o difuminado, es una operación usada en procesamiento de imágenes, generalmente para tomar los patrones de la imagen y eliminar el ruido. En el dominio de la frecuencia, el suavizado es un filtro pasa-bajas, donde las bajas frecuencias corresponden a áreas donde la intensidad varia lentamente (intensidad casi constante), mientras que las altas frecuencias son por cambios rápidos de intensidad. El suavizado también es de gran importancia cuando se desea reducir la resolución de una imagen.

cv::blur

Tiene como objetivo suavizar una imagen, remplazando cada pixel con el valor promedio de los pixeles vecinos que forman una ventana alrededor de éste. El tamaño de la ventana es variable y se determina con un parámetro de la función. En la figura 3–1 se observa una fotografía sin modificaciones. En la figura 3–2 se observa la fotografia original con un filtro cv::blur.

cv::GaussianBlur

Se usa cuando se quiere dar más peso en el promedio a los pixeles cercanos. Esto se logra usando un esquema que sigue una función gaussiana [4] ("una campana de gauss"). Los diferentes pesos pueden ser representados en una matriz que muestra los factores de multiplicación asociados a la posición de los pixeles vecinos, donde

Figure 3–1: Imagen original

el elemento central de la matriz corresponde al pixel donde se aplica el filtro. Esta matriz es llamada "Kernel" o mascara. En la figura3–3 se puede ver como el filtro GaussianBlur afecta a la imagen de la figura 3–1

Cv::GaussianBlur se usa cuando se quiere dar más peso en el promedio a los pixeles cercanos. Esto se logra usando un esquema que sigue una función gaussiana ("una campana de gauss"). Los diferentes pesos pueden ser representados en una matriz que muestra los factores de multiplicación asociados a la posición de los pixeles vecinos, donde el elemento central de la matriz corresponde al pixel donde se aplica el filtro. Esta matriz es llamada "Kernel" o máscara.

El filtrado gaussiano se realiza convolucionando cada pixel de la matriz de entrada con el Kernel gaussiano y luego haciendo la sumatoria para producir la matriz

Figure 3–2: Imagen aplicando un filtro cv::blur

de salida. Los valores del Kernel se asignan de acuerdo a la función gaussiana.

$$G(x) = Ae^{\frac{-x^2}{2\sigma^2}} \tag{3.1}$$

El coeficiente A se escoge de tal forma que el peso de la función sea 1. El valor de sigma controla el ancho de la campana. Entre más grande sea este valor, más plana es la campana. Para aplicar un filtro gaussiano 2D en una imagen es posible simplemente aplicar el filtro 1D en las filas y otro en las columnas. Esto es posible gracias a que el filtro gaussiano es un filtro separable.

cv::medianBlur

Es un filtro no lineal, por lo que no puede ser representado en una matriz o "kernel". Esta función remplaza cada pixel por el valor de la mediana de la ventana

Figure 3–3: Imagen aplicando un filtro cv::GaussianBlur

de pixeles vecinos. Es usado cuando se quiere eliminar ruido de puntos con valores muy altos que afectan considerablemente el promedio. en la figura 3–4 se tiene una imagen con ruido de sal y pimienta. Mediante este filtro se obtiene la imagen de la figura 3–5 donde se eliminan los pixeles con valores demasiado altos.

cv::bilateralFilter

Cuando se aplica un filtro blur o GaussianBlur se toma por hecho que los pixeles en una imagen real deben variar lentamente en el espacio, y así estar correlacionados entre ellos, mientras que el ruido varía abruptamente de un pixel a otro. Así, con estos métodos es posible eliminar el ruido, pero se pierden los bordes de la imagen. ver figura 3–6. bilateraFiler también es un promedio de los pixeles vecinos, pero el peso con el cual se computan tiene dos componentes. El primero es el mismo

Figure 3–4: Imagen con ruido sal y pimienta

usado en el suavizado gaussiano, que es la distancia al pixel del centro. El segundo también usa una función gaussiana pero no se basa en la distancia al centro sino en la diferencia de intensidad con el pixel del centro. Es decir que los pixeles con valores de intensidad cercanos tendrán más peso.

3.2.2 Detección de bordes

Los bordes son cambios abruptos de intensidad, esto es, que tienen una frecuencia alta. Aplicar un filtro pasa-altas amplificaría estas frecuencias. Como resultado, los filtros pasa-altas realizan detección de bordes. Existen varios métodos para la detección de bordes como; Operador Sobel, Canny, Prewitt, Roberts cross.

Filtro Sobel (Gradiente Sobel)

Es llamado un filtro direccional porque solo afecta las frecuencias horizontales o las frecuencias verticales, dependiendo del Kernel usado. EL operador Sobel se puede

Figure 3–5: Ruido Eliminado con medianBlur

usar para derivadas de cualquier orden. Las imágenes obtenidas mediante el filtrado horizontal y vertical se pueden unir para formar una aproximación del gradiente total de la imagen. ver figura3–7. Esto se puede lograr con una suma de ambas imágenes.

Canny

Es un operador que sirve para detectar bordes usando algoritmos basados en Sobel, aunque otros operadores pueden ser usados. A diferencia de otros métodos para detectar bordes como Sobel, los bordes de Canny son delgados y facilitan asignar valores de umbral que incluyan todos los bordes importantes y descarten los insignificantes. Además Canny trata de ensamblar los bordes individuales de la imagen para formar contornos, como se observa en la figura 3–8. Canny usa dos parámetros para los umbrales, uno inferior y el otro superior. El inferior se debe escoger de tal forma

Figure 3–6: Ejemplo de un suavizado con bilateralFilter

que todos los pixeles, que sean considerados que deben pertenecer al contorno de la imagen, sean incluidos.

3.2.3 Operadores Morfológicos

El filtrado morfológico transforma la imagen sondeándola con un elemento de forma predefinida. Las transformaciones morfológicas básicas son la dilatación y erosión, que son usadas en una gran variedad de contextos, como eliminar ruido, aislar elementos individuales, o unir elementos dispersos en la imagen. El instrumento fundamental es un "elemento estructurado" o "Kernel", es una configuración de pixeles (de cualquier forma y tamaño), que tiene definido un origen (punto ancla). Cuando se alinea el punto ancla con un pixel, la intersección del elemento estructurado con la imagen define un conjunto de pixeles sobre los que se aplica la operación morfológica específica.

Dilatación

Es un sondeo de una imagen con un elemento estructurado, normalmente un cuadrado o un disco con el punto ancla en el centro. A medida que el Kernel es escaneado sobre la imagen, se halla el pixel con el valor más alto de la intersección

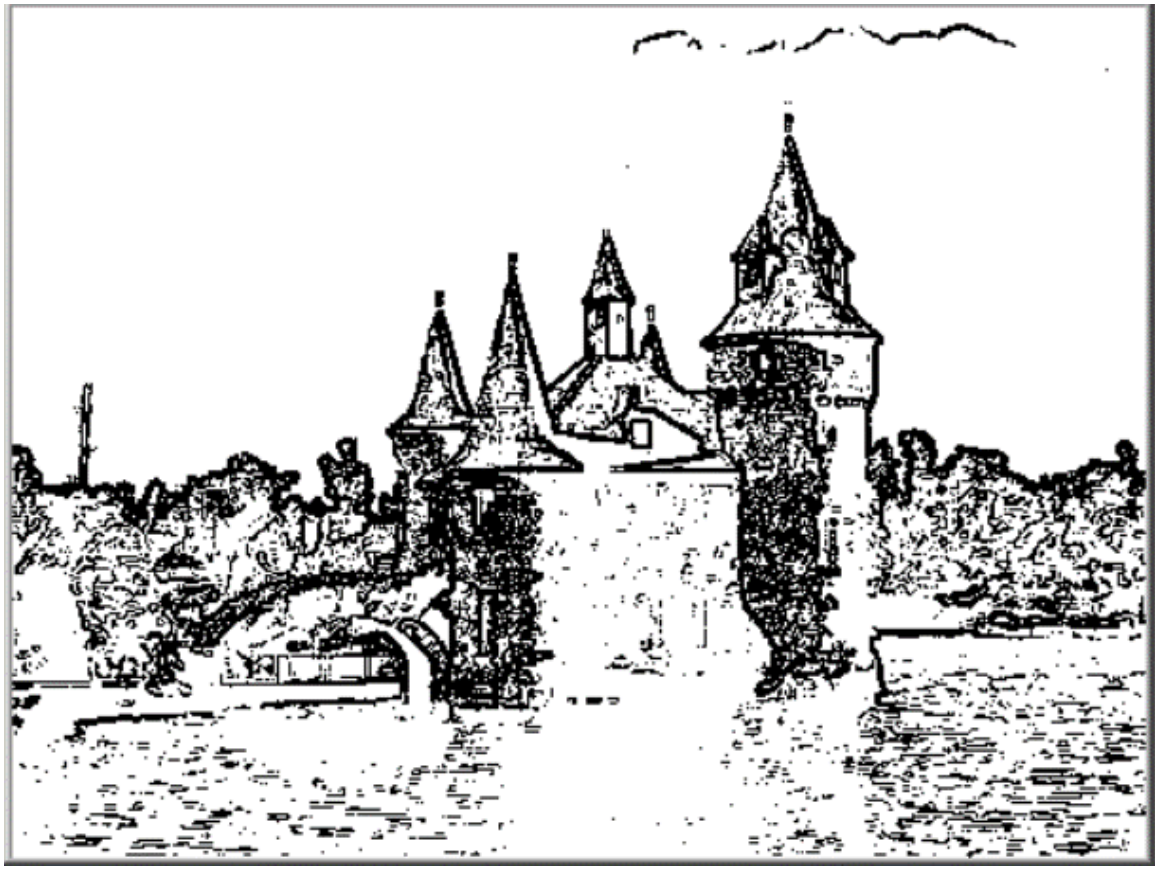

Figure 3–7: Gradiente Sobel de una imagen

y se remplaza el pixel sobre el que se encuentra el punto de ancla. Esto causa que las partes luminosas crezcan y por eso es llamado dilatación (ver figura 3–9).

Erosión

Hace un proceso equivalente, pero en vez de hallar el pixel con el mayor valor, se toma el mínimo valor dentro de la intersección (ver figura 3–10).

Opening y closing

Son combinaciones de los operadores dilatación y erosión. En el caso de *opening*, primero se erosiona y luego se dilata (figura 3–11. La función *closing* hace el proceso contrario, primero dilata y luego erosiona(figura 3–12. La función *closing* es muy usada cuando se quiere unir regiones, mientras que la función opening es usada cunado se quiere separar regiones.

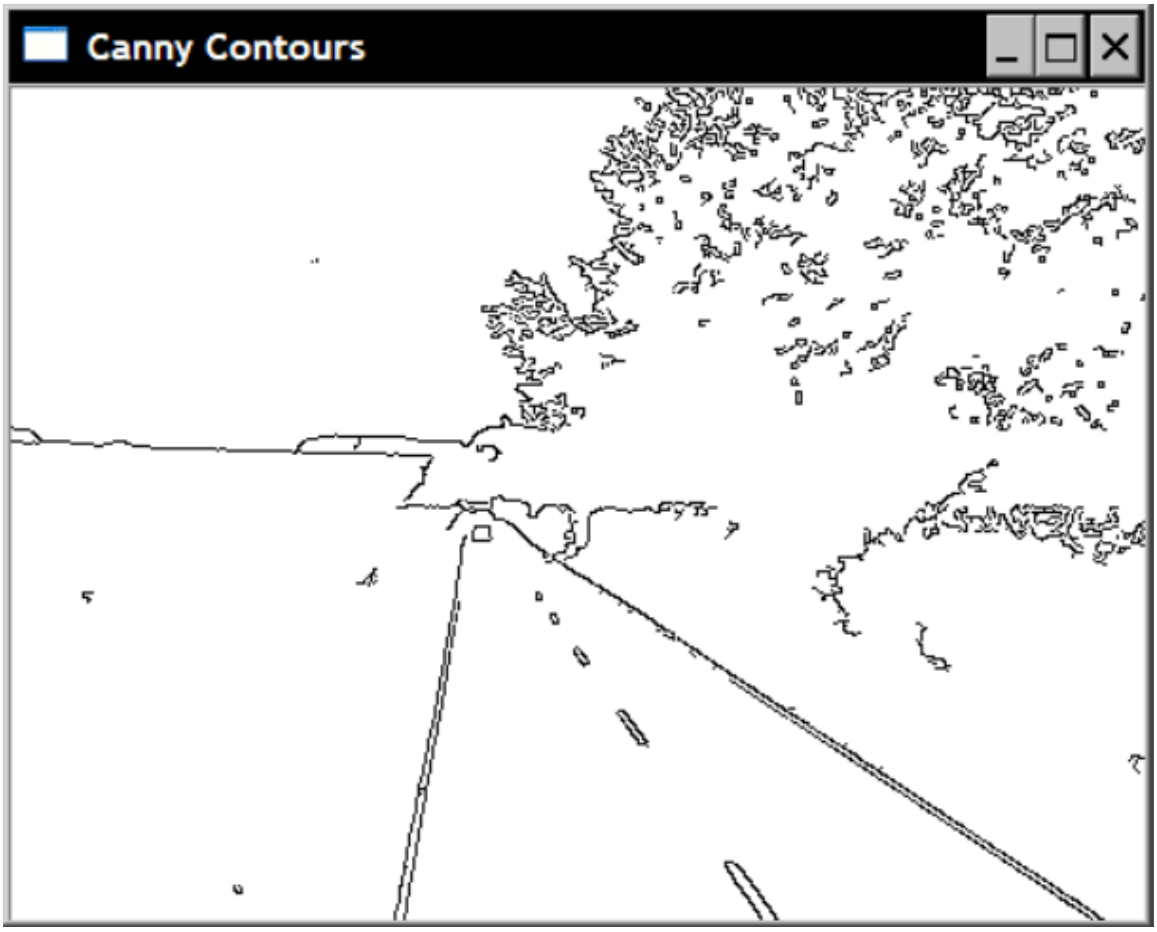

Figure 3–8: Bordes de una imagen obtenidos con la función Canny

Gradiente Morfológico

El gradiente morfológico se puede definir mediante la siguiente formula:

$$Gradiente(img) = dilatacion(img) - erosion(img)$$

Esta operación aplicada sobre una imagen booleana aislaría los perímetros de la imagen como se muestra en la figura 3–13.

3.2.4 Threshold

Cuando se quiere tomar una decisión sobre los pixeles de una imagen, para rechazar aquellos por debajo de un valor umbral y mantener a los demás. OpenCV ofrece una función cv::threshold que cumple con esta tarea. Hay varios tipos de *threshold*, y estos difieren en cómo se asignan los valores de los pixeles después de

Figure 3–9: Operación Dilatación

Figure 3–10: Operación Erosión

la comparación con el umbral. En la tabla3–1 se muestran los tipos de *threshold* y la asignación de valores para cada pixel. En la figura 3–14 se puede observar un ejemplo de *threshold* usando CV_THRESH_BINARY.

3.3 CÓDIGO DE BARRAS

Es una tecnología de identificación automática de uso mundial, que es usada para identificar y localizar los objetos a nivel comercial e industrial. El sistema consta de una serie de barras y espacios de distintos anchos que almacenan información que se denomina simbología y un conjunto de números impresos o código para su identificación por el hombre. Esta forma de identificación ha tenido gran aceptación por varias razones, por ejemplo su alta precisión, exactitud y confiabilidad, además

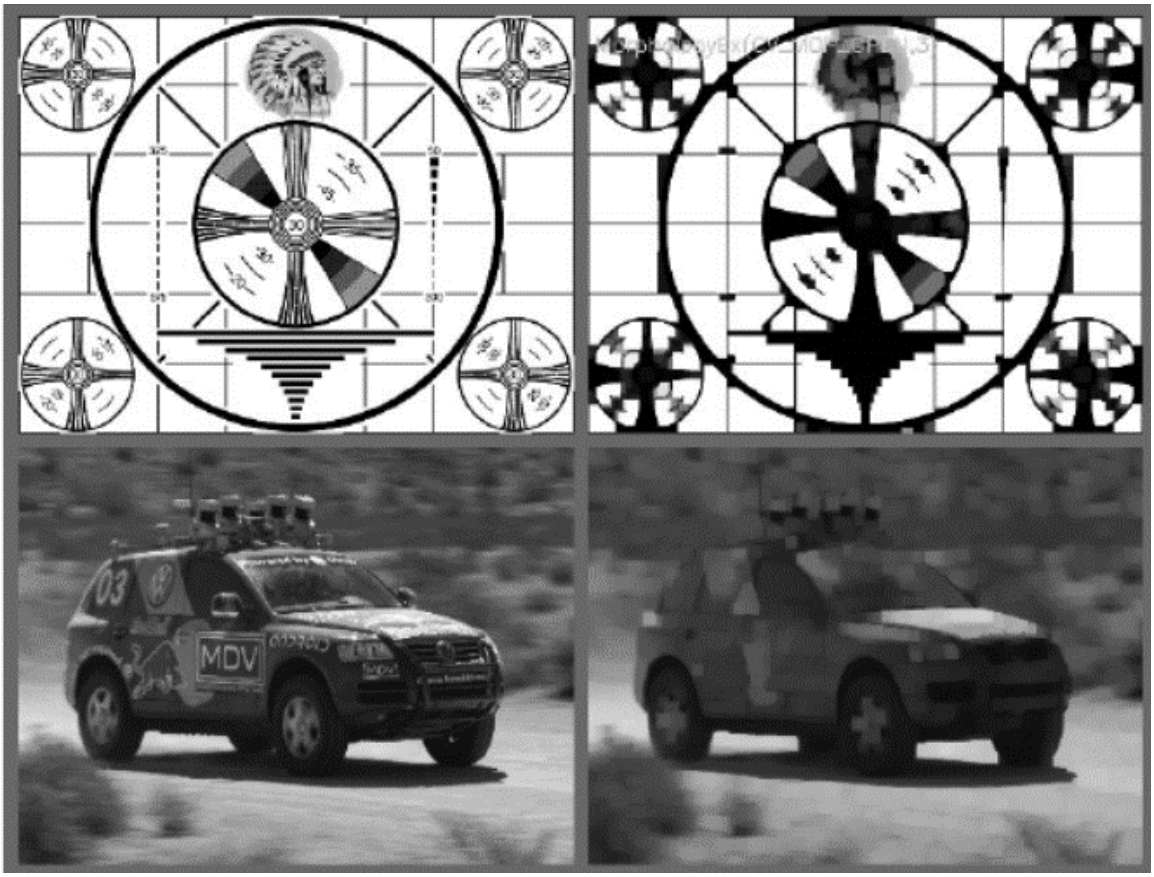

Figure 3–11: Operación Opening

Table 3–1: Opciones para la función Threshold

CV_THRESH_BINARY	dst =(src¿T)?M:0
CV_THRESH_BINARY_INV	dst =(src¿T)?0:M
CV_THRESH_TRUNC	dst =(src¿T)?M:src
CV_THRESH_TOZERO_INV	dst =(src¿T)?M:src
CV_THRESH_TOZARO	dst =(src¿T)?src:T

por su amplio portafolio de opciones para aplicaciones industriales, ya que es ideal para el manejo de productos a diferentes escalas. La asociación que se encarga de la asignación de estos códigos es la EAN (asociación internacional de numeración de artículos), la cual establece una identificación para cada país de dos o tres dígitos llamada FLAG. El industrial solicita a su asociación nacional, encargada de la asignación de códigos, un conjunto de números que identifica a su empresa y será único para todos sus productos, él podrá numerar cada uno de sus ítems definiendo así un código único. Cuando el industrial posee su conjunto de códigos debe diseñar los

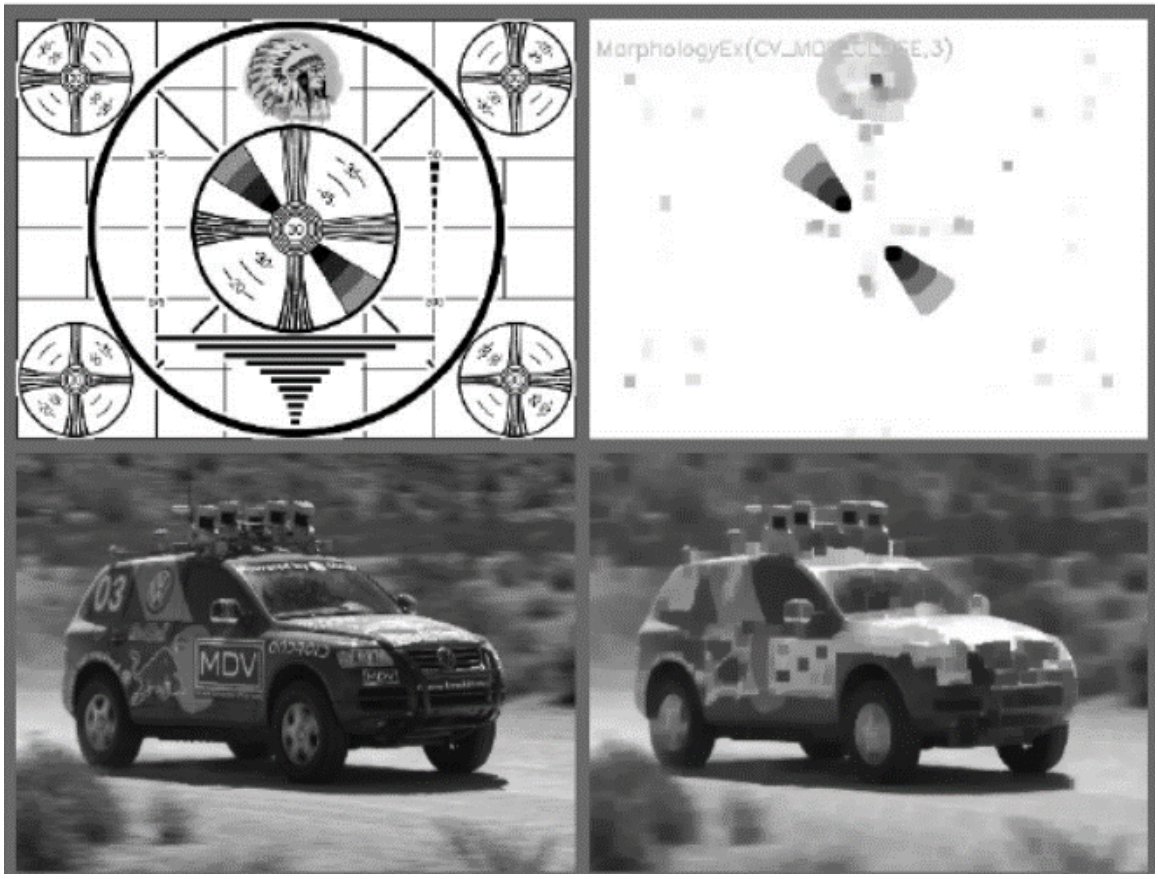

Figure 3–12: Operación Closing

empaques de tal forma que exista el espacio necesario para imprimir el código de barras según normas estrictas que son definidas por la asociación encargada, dependiendo del sistema utilizado, esto se aclarará posteriormente, el control de impresión es necesario ya que la implementación inadecuada de esta codificación puede impedir la comercialización automática del producto entre otros problemas.

El sistema de identificación por código de barras ofrece estos beneficios:

- Marcación única de cada producto desde su fabricación hasta el cliente final, el cual tiene la capacidad de conocer con precisión el precio del ítem que está comprando, evitando confusiones y posibles adulteraciones.
- Facilidad de control en los procesos de producción, almacenamiento, transporte y venta.

Figure 3–13: Detección de bordes con gradientes morfologicos

- Con la información almacenada en el código (simbología) se pueden establecer estadísticas que permitan conocer comportamientos de los productos en el tiempo.
- Cada sistema de codificación posee un método de autoverificación que reduce al mínimo los errores en las lecturas.
- Teniendo en cuenta posibles daños en el código impreso la medida vertical se sobredimensiona de tal forma que con un 5% de este valor se logre satisfactoriamente la medida.
- Ofrece velocidad y eficiencia especialmente en la ventas al consumidor, en las cajas de los supermercados.
- Se elimina la necesidad de remarcado de precios producto por producto, cuando sea necesaria alguna variación en este valor.

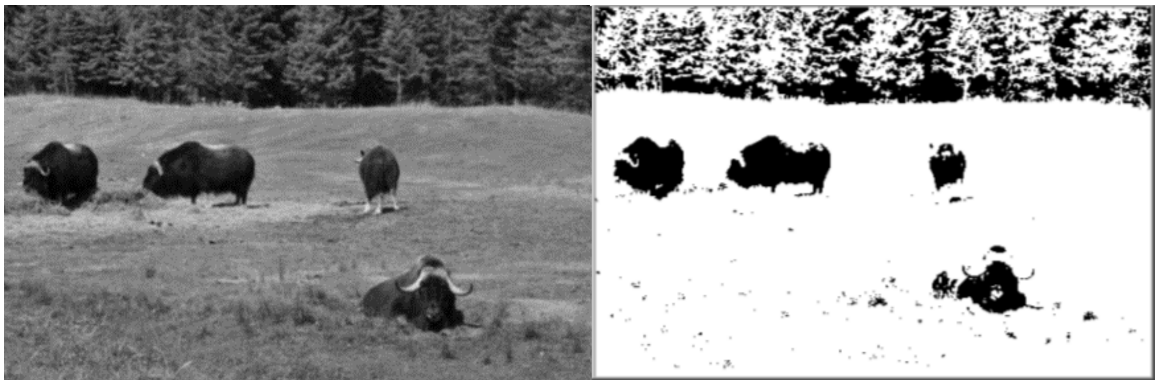

Figure 3–14: Binarización de una imagen usando Threshold

- Mejor manejo de inventarios por tener información disponible en tiempo real.
- Adaptación a la mayoría de los sistemas de embalaje, impresión y materiales de envases disponibles en la industria.

Existen diferentes sistemas de codificación por barras, que difieren en la cantidad de caracteres, en los países utilizados y en las normas aplicadas para el diseño, como son EAN 13, EAN 8, UPC A, UPC E, entre otros. Actualmente aquellos son los únicos códigos de barras permitidos para productos escaneados en puntos de venta minoristas[1] .

3.3.1 EAN 13

Consta de una cantidad fija de 30 barras y 29 espacios que codifican la información, teóricamente permite clasificar 1000 países, para cada uno de ellos 10000 industrias y en cada una de estas 10000 productos o presentaciones. El código EAN 13 precisa trece caracteres (ver figura 3–15), doce de los trece se representan por dos barras y dos espacios ubicados alternativamente, es decir, que existen cuatro elementos para cada carácter, el otro no se representa de esta manera. Una representación y otra se diferencian por el ancho y ubicación de los elementos. Un módulo es el elemento más angosto, sea barra o espacio en un código de barras, todos los elementos

[1] http://www.gs1us.org/resources/standards/ean-upc

del código poseen un ancho, múltiplo del módulo. El ancho de cada carácter es fijo y mide 7 módulos (ver tabla 3–3), por lo tanto los cuatro elementos que forman un carácter tendrán un ancho de siete módulos. Existen tres formas de codificar los caracteres numéricos que se representan con barras y espacios, estas formas se denominan "A, B, C". Los tipo A tienen en total dos barras formadas por tres o cinco módulos (impar), el primer módulo a la izquierda es un espacio y el último es una barra, se ubican a la izquierda del separador central. Los caracteres que pertenecen a la clasificación B están formados por dos barras en total, compuestas por dos o cuatro módulos (par) el resto de características son idénticas a los tipo A, incluyendo la ubicación. Los tipo C son de la forma par, el primer módulo a la izquierda es una barra y el último es un espacio y su ubicación es exclusivamente después de los separadores centrales. Para determinar el carácter que no se representa con barras y espacios se revisa la secuencia de los caracteres ubicados en la parte izquierda de los separadores centrales (tipos A y B), el valor numérico va definido por las secuencias que se muestran en la tabla 3–2

Table 3–2: Posiciones de los seis caracteres de la izquierda de los separadores centrales

Valor numérico definido	Secuencia de caracteres tipo A y B
0	A A A A A A
1	A A B A B B
2	A A B B A B
3	A A B B B A
4	A B A A B B
5	A B B A A B
6	A B B B A A
7	A B A B A B
8	A B A B B A
9	A B B A B A

Existe un carácter de verificación o control que se ubica en la parte derecha del código, este digito es el resultado de un cálculo en el que intervienen todos los demás

números, con excepción del símbolo que no se representa con barras y espacios, la función de este carácter de control es evitar errores de lectura y detectar estos errores, como característica especial este digito permite su autoverificación, es decir que está en la capacidad de reconocer un error en su propio cálculo de asignación. El método para calcular el valor de este carácter consiste en: Multiplicar el valor de cada carácter en posición impar x 1, multiplicar los de posición par x 3, obteniendo 12 resultados entre 0 y 27. Los factores 1 y 3 reciben el nombre de ponderación. Luego se suman los 12 productos. Esta respuesta se llama suma de productos, la cual se divide entre 10 obteniendo un cociente y un residuo. Finalmente se le resta a 10 el residuo y la respuesta arrojada es el valor del carácter de verificación.

Figure 3–15: Código de barras EAN 13

Table 3–3: Patrones de dígitos para símbolos

Number	Left Digits		Right Digits
	Odd Parity	Even Parity	
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

3.3.2 EAN 8

Esta versión del sistema EAN se utiliza exclusivamente cuando el tamaño o la forma del producto no disponen de espacio suficiente para imprimir el código EAN 13, debido a que consta de tan solo 8 dígitos (ver figura 3–16). EAN 8 no solo se diferencia por el número de dígitos con EAN 13, ya que la información almacenada es independiente una de la otra. Esta codificación tiene algunas desventajas con respecto a EAN 13, por ejemplo la capacidad de codificación más limitada y para usar este sistema debe acudir a la institución local de codificación.

Figure 3–16: Código de barras EAN 8

3.3.3 UPC A y UPC E

Al igual que en las simbologías de EAN 13, UPC A codifica cada carácter mediante dos barras y dos espacios, y tiene un ancho fijo de 7 módulos, pero el código tiene una longitud de 12 caracteres, que se asignan de la siguiente forma: Carácter 12 (digito más a la izquierda), categoría del producto (medicinal, alimenticio, etc). Caracteres del 11 al 7, identificación del fabricante. Caracteres del 6 al 2, identificación del producto. Carácter 1, digito de verificación. La longitud de las barras que representan los dígitos 1 y 12 es más larga que las demás. El código UPC E, también conocido como código reducido es una versión en la que se eliminan por lo menos 4 ceros en el código. Existen cuatro formas de suprimir los ceros, por lo

que no deben eliminarse de forma aleatoria. Estos códigos solo se usan en Canadá, Estados Unidos, Australia, Nueva Zelanda y Reino Unido.

3.3.4 codificación internacional de libros (ISBN) y revistas (ISSN)

La numeración de libros publicados en todo el mundo está a cargo de la representación de cada del ISBN (International Standard Book Numbering System). Esta codificación utiliza el sistema EAN quien asigno los flags 978 y 979 a la agencia internacional, de esta forma el código ISBN consta los tres dígitos de la izquierda que representan los flags mencionados anteriormente, los siguientes nueve números identifican a cada uno de los libros, y el último carácter de la derecha es el de verificación o control. La agencia encargada de la codificación de las revistas ISSN (International Standard Serial Numbering System), de igual forma que con los libros, las codifica utilizando EAN que asignó el flag 977 para este propósito, de tal forma que la composición completa del código consta de tres dígitos del flag, siete para la identificación de la publicación, dos dígitos extras y el último carácter de verificación.

3.3.5 diseño de las dimensiones del código para los sitemas EAN Y UPC

Para poder seleccionar el tamaño del código de barras en el empaque, es necesario tener en cuenta varios factores que pueden afectar de forma directa o indirecta la lectura de la información. Estos factores son los siguientes: Dimensiones del código, teniendo en cuenta el factor de magnificación, el cual será descrito posteriormente. La forma que tiene el envase en donde se va a imprimir el código. El sistema de impresión que se va a implementar. El sentido de avance de la impresora. La disponibilidad real de espacio y factores económicos. Para entender el factor de magnificación se definirá como tamaño normal las dimensiones standard de un símbolo (4 cm de ancho)14, el cual tendrá como valor del factor de magnificación 1, según las normas este factor no debe tener valores por fuera del intervalo de 0.8 y 2.0, esto equivale al 80 y 200% del valor standard, esto con el fin de disminuir

posibles errores. Al tener en cuenta la forma del envase, también se piensa en la superficie en la cual se imprimirá el código, la cual se prefiere plana, porque en el caso contrario es necesario calcular la curvatura. Estas limitaciones también definirán el factor de magnificación que se deberá emplear.

3.3.6 Escaner de códigos de barras asistido

Este dispositivo tiene como función leer la simbología del código de barras moviendo una pieza móvil a través de barras y los espacios, el otro elemento que compone al lector es el decodificador, quien recibe la señal del dispositivo de entrada y devuelve el dato para manejarlo de la forma en que se necesite. El dispositivo (figura 3–17 de entrada, que es móvil determina la localización del código de barras dirigiendo una luz en dirección a las barras, y determinando que cantidad de esa luz es reflejada. La salida de este elemento puede ser un voltaje analógico o un voltaje digital. El decodificador hace el papel de computador, recibe la salida del dispositivo móvil y procesa esta información para así obtener la simbología del código. El proceso de decodificación está compuesto por varios sub-procesos, que se describen a continuación: Inicialmente se debe definir si la información recibida por el dispositivo móvil pertenece a una barra o a un espacio, esto se hace generalmente comparando la señal con un umbral determinado por un histórico de lecturas recientes. El siguiente paso consiste en medir el ancho de cada elemento del código, definiendo el módulo. Para poder decodificar la simbología se compara cada carácter con una tabla establecida, teniendo en cuenta el valor del ancho del módulo calculado anteriormente. Dependiendo de la dirección de escaneo es necesario o no invertir la información obtenida. Si la lectura se hizo de derecha a izquierda es necesario hacer este cambio. Para determinar la dirección se examina patrones de inicio y final de códigos. Posteriormente se verifica la validez de la información obtenida. Finalmente se transmite la información decodificada al siguiente paso del proceso en el que se

está trabajando. Existen tres grandes categorías en las que se pueden agrupar a los decodificadores, on-line, portables y portables on-line.

Figure 3–17: Escaner de código de barras

3.4 CÁMARA FOTOGRÁFICA

Una cámara está formada por mecanismos que tienen como función adquirir la imagen reflejada por los objetos, y proyectarlo sobre una película impregnada de una sustancia fotosensible, de tal forma que sobre cada punto de la película incida la luz proveniente de un cono visual tan estrecho como sea posible[2] . A la hora de tomar fotografías es necesario tener en cuenta algunos factores que afectan la adquisición de una imagen, por ejemplo, si el objeto que se va a fotografiar está en movimiento, la configuración de la cámara es diferente que si el mismo objeto está estático, de igual forma es necesario cariar parámetros dependiendo de la distancia, la intensidad de luz etc. Los principales ajustes que se deben tener en cuento son: Enfoque: al variar esta propiedad realmente se está manipulando la distancia focal que es el trecho que existe entre el centro óptico del lente y el foco (punto focal). El foco es

[2] http://mecfunnet.faii.etsii.upm.es/difraccion/camera.html

un punto donde se concentran los rayos de luz. El objetivo de la cámara es la medida que se refiere a la distancia entre el diafragma y el foco. Comúnmente se conoce a la alteración de este parámetro como ZOOM, ya que con l conseguimos un mayor o menor acercamiento al objeto. En la actualidad la mayoría de las cámaras poseen sistemas autofocos como los siguientes[3]

1. Comparación de contrastes: es el sistema más común. Su funcionamiento consiste en un panel fotosensible que recoge dos imágenes, una procedente del visor y otra de un espejo móvil acoplado al motor de enfoque. El objetivo comienza a enfocar desde el infinito y detiene el motor cuando el contraste entre luces y sombras coincide en las dos imágenes18.
2. Infrarrojo: este sistema se basa en un rayo infrarrojo que la cámara dirige al objeto, la reflexión de dicho rayo es recolectada por un espejo que detiene el proceso cuando detecta una señal de intensidad máxima. Este procedimiento no se ve afectado por superficies poco refractivas.
3. Ultrasonidos: su funcionamiento es similar al método de rayo infrarrojo, pero en este caso se utiliza una señal de audio inaudible. Un cronometro identifica el tiempo que se demora en regresar la señal de audio, con este dato se calcula la distancia.

Tiempo de exposición: el tiempo que el obturador permite que la luz entre en la cámara se denomina, tiempo de exposición. En el caso de seleccionar un tiempo de exposición demasiado largo se produce un efecto que se denomina trepidación, y que muestra la imagen en movimiento. Contraste: es la diferencia que existe entre las zonas oscuras y claras de la foto. Se define un contraste alto cuando hay muchas luces y sombras presentes en la imagen, y contraste bajo cuando no una diferencia

[3] http://www.difo.uah.es/curso/la_camara_fotografica.html

marcado entre zonas oscuras y claras[4] . Ganancia: es la capacidad que posee la cámara de reforzar electrónicamente la iluminación de la imagen en casos de escasa luz. Saturación: la saturación de un color es el inverso de la cantidad de gris que contiene. Cuanto más alto sea el contenido gris, menor será la saturación. En otras palabras en el formato RGB, la saturación es el grado en que uno de estos tres componentes predomina, de tal modo que a medida que se igualan estos valores el color va perdiendo saturación hasta llegar a un gris o blanco[5] .

3.4.1 Partes de la cámara

Las principales partes de una cámara de fotos son el objetivo, la película, el obturador y el diafragma. Objetivo: está compuesto por un lente o un conjunto de lentes que pueden ser fijos o móviles los cuales tienen como función formar una imagen del objeto sobre la película. Película: es un material sensible a la luz que reacciona químicamente cuando es iluminado, esta variación guarda la información visual en cada fotografía. Obturador: este dispositivo controla el paso de luz a la película, modificando el tiempo de apertura. Estos tiempos se miden en la escala de las centésimas de segundo. Diafragma: está compuesto por un conjunto de láminas finas que tienen la función de estrechar el paso de luz que entra a la cámara.

3.4.2 Tipos de cámara

Las cámaras se pueden clasificar según diferentes criterios, por ejemplo tipo de almacenamiento de imágenes (analógico o digital), tamaño o por el grado de automatismo (esto solo aplica a las cámaras analógicas que pueden ser manuales o automáticas). En este documento se enfatiza en las cámaras digitales ya que una de este tipo es la que se utiliza en el desarrollo de este proyecto.

---

[4] http://fotonovata.wordpress.com/2008/06/26/el-contraste-en-la-fotografia/

[5] http://www.fotonostra.com/glosario/saturacolor.htm

3.4.3 Tipos de cámara según tamaño

Cámaras WEB: son cámaras digitales pequeñas que no posees visor y que necesitan ser conectadas a un computador para poder visualizar las imágenes captadas. Su funcionamiento se basa en un filtro RGB quien recibe la luz que pasa por la lente y entrega los valores de sus componentes rojos, verde y azul. Estos componentes se concentran en un dispositivo sensible a la luz denominado CCD (Charged Couple Device), el cual asigna valores binarios a cada pixel, esta información queda disponible para ser utilizada en procesos posteriores[6] . Cámaras compactas: se caracterizan por su objetivo que no es desmontable, por tener el sensor digital que capta la imagen muy pequeño, lo cual limita la calidad de las fotos considerablemente. Intermedias o bridge: ofrecen mejor calidad en las imágenes tomadas debido a que el tamaño del sensor es más grande, aunque su objetivo no es intercambiable, el control sobre la obtención de la imagen es bueno. Réflex o DSLR: el tamaño del sensor es considerablemente mayor comparándolo con las categorías anteriormente mencionadas, y teniendo en cuenta que la relación señal a ruido es directamente proporcional a esta dimensión, ofrece mayor nitidez y calidad en las fotos. Como principales ventajas permite el intercambio de objetivos, dispone de un visor réflex que muestra con mucha precisión el resultado definitivo de las fotos y permite manejar con más exactitud el proceso de tomar una foto. La principal diferencia entre una cámara réflex y una compacta, es que la primera permite un control completo de la profundidad de campo, lo que permite seleccionar el objeto de forma precisa y nítida en cualquier ambiente. Medio formato: son dispositivos con la capacidad de tomar fotos de alta calidad, su aplicación está limitada a la fotografía profesional y al ámbito científico.

---

[6] http://www.informaticamoderna.com/Camara_web.htm

3.5 UBUNTU

Ubuntu es un operativo enfocado en simplificar su instalación y uso, dar libertad al usuario y tener constantes actualizaciones, generalmente cada 6 meses. Es un software libre y gratuito que posee en entorno de escritorio propio llamado Unity(ver figura 3–18).

Ubuntu posee aplicaciones orientadas al usuario, las cuales tienen como función facilitar el desarrollo de actividades. Las principales aplicaciones que el sistema trae por defecto son: Mozilla Firefox como navegador web, la mensajería se maneja con Empathy, cliente de correo Thunderbird, lector de documentos PDF Evince, entre otras. Algunas de las características que hacen a este sistema operativo el más popular de las distribuciones de Linux[7] son: la capacidad de soportar arquitecturas de 32 y 64 bits además se puede implementar en otros dispositivos que tienen arquitecturas como ARM, PowerPC, SPARK, etc. El diseño de la interfaz de usuario se identifica por ser organizado y de fácil entendimiento, esta interfaz se compone por la barra superior para indicadores de sistema, el lanzador de aplicaciones al lado izquierdo y el tablero que despliega los accesos a aplicaciones y medios. Además el sistema incluye funciones avanzadas de seguridad.

3.6 GUVCView

Este sofwuare es un GTK+ (http://www.gtk.org/) que está enfocado en la captura de imagen y video utilizando los dispositivos que sean soportados por un UVC[8] . GUVCView se diseñó basado en el proyecto luvcview[9] el cual fue creado por

---

[7] http://paolabocaranda.wordpress.com

[8] (http://www.logitech.com/es-es/webcam-communications/video-software-services/articles/4770)

[9] (http://www.marlonj.com/blog/2009/02/luvcview-visor-para-webcams-en-ubuntu-810/)

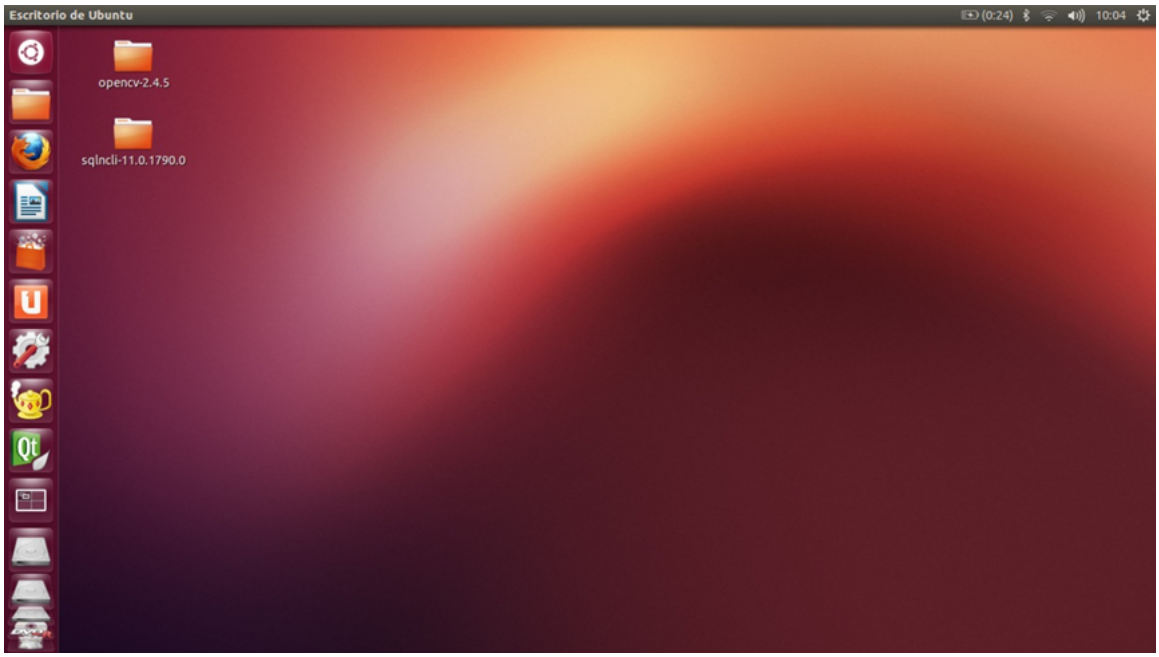

Figure 3–18: Escritorio de Ubuntu

el equipo QuickCam. Las principales características de GUVCView son la simplicidad para capturar imágenes, videos o imágenes con video de alta calidad, la posibilidad de controlar varios factores que afectan la captura como brillo, intensidad, enfoque, entre otros (ver figura 3–19), usando herramientas simples como barras deslizables y cajas de chequeo[10] .

3.7 Qt

Qt es un framework multiplataforma que es comúnmente utilizado para desarrollar aplicaciones que tengan o no interfaz gráfica de usuario, en estas GUI es posible usar herramientas como botones, cajas de texto (de entrada o salida), barras deslizables, entre otras. Qt se desarrolla bajo el concepto de software libre y código abierto a través de Qt Project, en donde participan la comunidad en general y grupos de

[10] http://www.ghacks.net/2011/02/05/record-from-your-web-cam-in-linux-with-guvcview/

Figure 3–19: Captura de pantalla de Guvcview

desarrolladores de empresas como Nokia y Digia. El lenguaje nativo para trabajar en Qt es C++, pero se pueden aplicar otros lenguajes a través de blindings[11] .

Qt creator es un entorno de desarrollo integrado IDE creado por Trolltech para el desarrollo de aplicaciones con Qt en sus versiones superiores a la 4.0, está disponible en las plataformas Linux, Mac y Windows (ver figura3–20).

3.8 ZBar

ZBar es un software libre de código abierto que tiene como objetivo leer los códigos de barras desde varias fuentes. Este programa suporta varios sistemas de

[11] http://softpei.blogspot.com/2013/05/introduccion-al-framework-qt-5-para-el.html

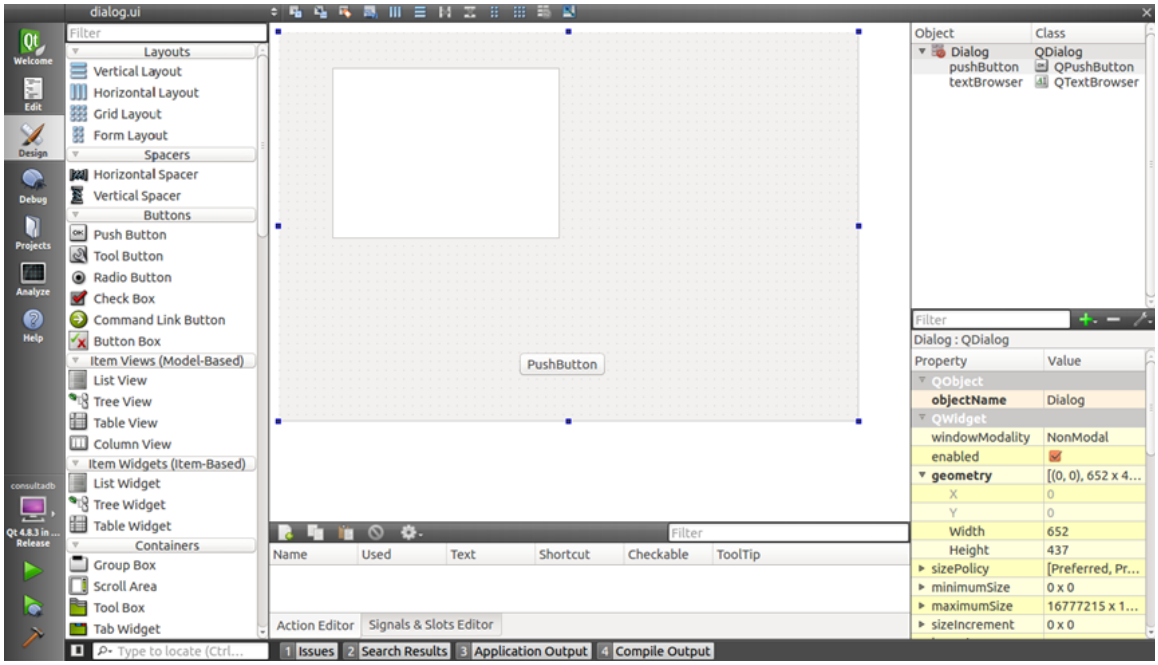

Figure 3–20: Captura de pantalla de Qt creator

codificación, entre ellos EAN-13, UPC y QR, que son las simbologías más comunes en el mercado. ZBar se caracteriza por tomar las lecturas de códigos rápidamente, y por la capacidad de ser integrado en varias arquitecturas, incluyendo sistemas de dispositivos móviles. El software es ejecutable con interfaz gráfica y sin ella, esto permite ampliar las posibles aplicaciones de uso de este programa[12] .

Principales características

- Multiplataforma.
- Alta velocidad de escaneo.
- Demanda poca memoria.
- No tiene límite en el número de imágenes.
- Se compuesto de componentes modulares que pueden ser usados en conjunto o por separado.

[12] http://zbar.sourceforge.net/

3.9 SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje de datos normalizado, que se utiliza para crear objetos QueryDef, y con el método Execute se puede crear y manipular directamente las bases de datos Jet y crear consultas SQL de paso para manipular bases de datos remotas cliente servidor. Este lenguaje está compuesto por comandos, cláusulas, operadores y funciones de agregado. Una de las principales funciones de este lenguaje es hacer consultas de selección, las cuales se utilizan para indicar al motor de datos que extraiga y entregue información de una base de datos[13] . Algunos de los sistemas de gestión de base de datos con soporte SQL más utilizados son:

3.9.1 Microsoft SQL Server

Es el sistema de gestión de base de datos creado por Microsoft en 1989, y des de entonces han generado alrededor de 10 versiones.

3.9.2 MySQL

Es otro sistema de gestión de base de datos desarrollado por Sun Microsystems, es muy popular y se caracteriza por ser relacional y multiusuario.

3.9.3 SQLite

Sistema de gestión de base de datos relacional que está contenida en una pequeña biblioteca escrita en C.

PostgreSQL

Sistema de gestión de bases de datos objeto-relacional con código fuente libre.

3.10 OCR

El reconocimiento óptico de caracteres (OCR) es una tecnología dirigida a la digitalización de textos que estén contenidos en imágenes. Al analizar una imagen

[13] http://www.unalmed.edu.co

con texto, inicialmente se identifican los caracteres, para posteriormente almacenar estos símbolos en forma de datos. Una aplicación en la que se usa frecuentemente este proceso de reconocimiento es la conversión de documentos escaneados o en formato PDF en archivos editables.

3.11 SISTEMAS PARA DESPACHO

3.11.1 Sistema de banda central (Central Belt System) de la empresa Knapp AG

Este sistema consiste en un conjunto de automáticos que están dispuestos a lo largo de una banda transportadora central, cada uno posee un tipo de producto, es cual es lanzado a la banda por medio de un eyector. La selección de medicamentos se hace automáticamente de acuerdo a la orden solicitada. La banda transportadora dirige los productos a un punto de entrega en donde el pedido se empaca en cajas [14] . Ver figura 3–21. Es posible que se presenten problemas con los productos que tienen cuna forma irregular, esto es muy común en la industria farmacéutica, ya que los eyectores convencionales pueden no tener un rendimiento óptimo en estos casos, esto se puede solucionar implementando el eyector de productos universal (Universal Product Dispenser UPD). Ver figura 3–22

Existe un sistema similar que está dedicado a los productos con envases cilíndricos, se llama "Canal de tubos TKA" (ver figura3–23).

[14] http://www.knapp.com/cms/cms.php?pageName=glossary&iD=70

Figure 3–21: Sistema de banda central

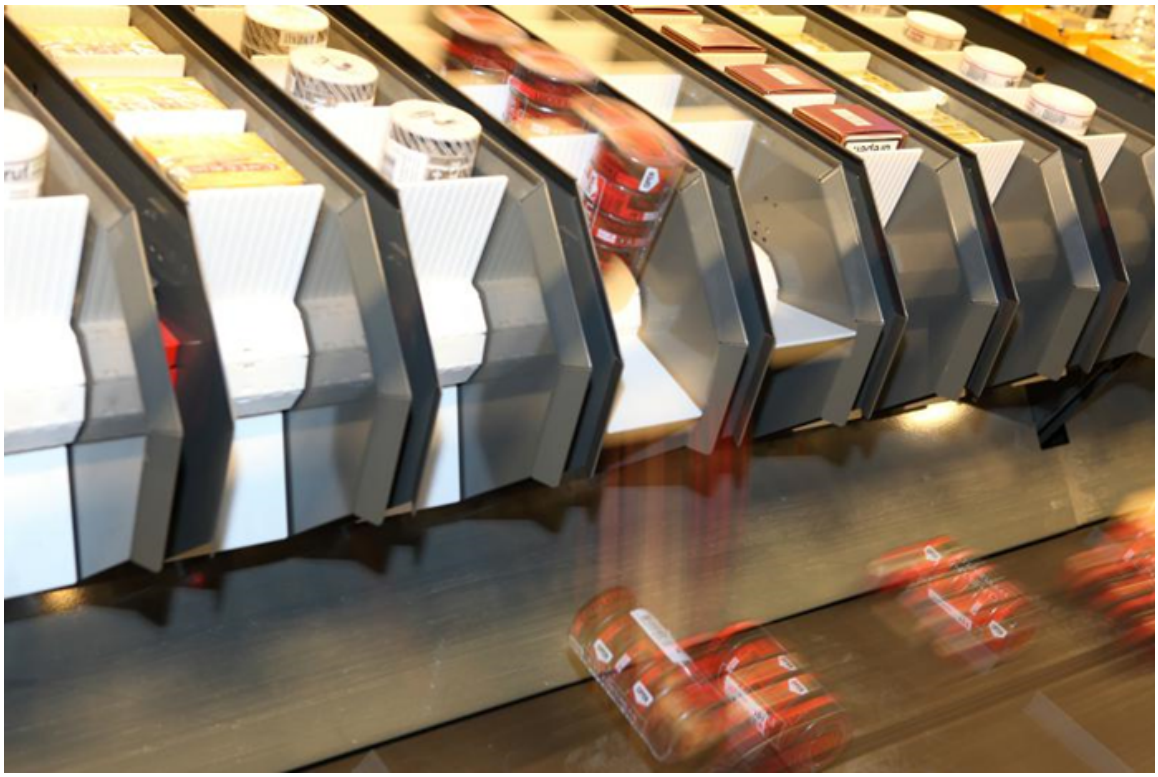

Figure 3–22: Eyector de productos universal

Figure 3–23: Eyector de productos cilíndricos

CAPITULO 4
PLANTEAMIENTO DEL SISTEMA

Para verificar los despachos se seleccionó leer el código de barras de cada ítem. Esto con el fin de tener la posibilidad de comparar la cantidad escogida con la facturada. El sistema desarrollado usa una banda transportadora donde un operario debe colocar cada medicamento, de tal forma que el código pueda ser captado. Con los códigos de barras se puede llevar un conteo de los productos que están siendo despachados y así verificar la exactitud de la entrega.

4.1 SELECCIÓN DE DISPOSITIVOS

Al seleccionar los códigos de barras como medio de identificación, se tiene la opción de escoger entre dos dispositivos de entrada, un lector de códigos o una cámara digital. El lector es un mecanismo especializado solo en esta tarea, mientras la cámara puede cumplir esta función además de otras. Por esta razón se prefirió trabajar con la cámara digital, ya que con ella es posible implementar mejoras al sistema en un futuro, por ejemplo la determinación del estado de los productos por medio de procesamiento de imágenes.

4.1.1 Selección de la cámara

Para esta aplicación se requiere una cámara con una resolución suficiente tal que pueda definir los bordes de los códigos más pequeños a una distancia determinada fija, teniendo en cuenta que una resolución demasiado grande aumentaría el tiempo de procesamiento de la imagen, por esto se consideró un tamaño en un intervalo de 1 a 2 megapíxeles, estos valores se calcularon a partir de pruebas realizados

con imágenes a diferentes resoluciones. Para una óptima comunicación entre la cámara y el computador, se requiere que la transferencia de datos sea USB; por esta razón una cámara UVC es ideal para esta aplicación. Debido a que el medicamento está en constante movimiento por estar sobre la banda transportadora, es necesario tener la capacidad de modificar parámetros de la cámara como brillo, saturación, foco, ganancia, contraste y exposición con el fin de evitar el efecto de trepidación. Teniendo en cuenta los requerimientos mencionados anteriormente se escogió la cámara Logitech QuickCam Pro 9000, que tiene las características en la tabla 4–1:

4.1.2 Banda Transportadora y variador de velocidad

La banda transportadora fue provista por Offimedicas S.A. Cuenta con un variador de frecuencia marca Danfoss, que controla la velocidad del motor de una potencia de 0,5 hp, que la mueve.

4.1.3 Iluminación

Debido a que la imagen se debe captar en movimiento, el tiempo de exposición tiene que ser muy corto. Esto implica que la iluminación necesaria para que la imagen sea clara debe ser abundante y directiva. Para cumplir con estos requisitos se escogieron un par de luces led que se pueden dirigir según la necesidad. Se descartó alguna lámpara fluorescente porque con este tipo de iluminación se presenta un efecto de parpadeo que afecta a la imagen, ya que no da luz continua. También se descartaron las lámparas incandescentes porque al estar encendidas disipan mucha energía en forma de calor.

4.2 DESARROLLO DEL PROGRAMA

4.2.1 Configuración de la cámara

Para asegurar que el funcionamiento del sistema sea el deseado, se debe fijar la configuración de la cámara, de tal forma que siempre que se inicie el proceso de verificación, los parámetros de ésta mantengan los valores definidos. Estos parámetros

Table 4–1: Características de la cámara

Camera Specifications:	
Connection Type	Corded USB
USB Type	High Speed USB 2.0, UVC
USB VID_PID	VID_046D&PID_0990
Microphone	Built-in, Noise Cancellation
Lens and Sensor Type	Glass, CMOS
Focus Type	Auto
Field of View (FOV)	75 Diagonal FOV
Focal Length	2 mm
Optical Resolution (True)	2 MP (1600x1200)
Image Capture (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (JPG - True) 3 MP (2048x1536), 4 MP (2304x1728), 5 MP (2592x1944), 8 MP (3264x2448) (JPG - Software Enhanced)
Image Capture (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (JPG - True)
Video Capture (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (WMV - True)
Video Capture (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (WMV - True)
Frame Rate (max)	30 fps @ 800x600, 15 fps @ HD 960x720 (Hardware Limit)
Video Effects (VFX)	N/A
Right Light	Right Light 2
Buttons	Snapshot
Indicator Lights (LED)	Activity/Power
Privacy Shade	Software (NOTE:Mutes video but not audio)
Clip Size (max)	6.35 mm or .25 inch to infinity (Not Detachable)
Cable Length	6 Feet or 2 Meters

se controlan con un driver dedicado a cámaras UVC, este software se llama GUVCVIEW, con él se controlan los valores del brillo, distancia focal, nitidez, contraste, exposición y ganancia. Una vez definida la configuración, se puede guardar en un archivo con extensión gpfl el cual puede ser llamado desde el código principal, para ajustar los parámetros de la cámara cada vez que se inicie el sistema. El código que se usó para llamar el archivo que contiene la configuración fue:

```
int status = system("guvcview -l config_camara.gpfl");
```

System es la instrucción que permite ejecutar un comando desde la terminal en sistemas operativos Linux, en este caso se ejecuta el controlador de la cámara con una configuración almacenada anteriormente. Por otra parte la resolución de las imágenes capturadas se define en el código principal, ya que el controlador de la cámara no tiene esa opción. El tamaño de la imagen se especifica con las siguientes líneas de código:

```
cap.set(CV_CAP_PROP_FRAME_WIDTH,1600);
cap.set(CV_CAP_PROP_FRAME_HEIGHT,1200);
```

4.2.2 Detección de movimiento

La captura de las imágenes se inicia simultáneamente con el sistema, pero se tiene que cumplir alguna condición para empezar a guardar datos, ya que no es necesario guardar imágenes que no contengan información útil. Por esta razón solo se almacenarán los datos de las imágenes en las que los productos estén presentes. Para asegurar que la información almacenada sea útil, es necesario que solo se guarden las fotos donde esté presente el medicamento, por esto solo se salvan datos cuando se detecta que el producto está frente a la cámara. Para detectar la presencia, se detecta movimiento utilizando un método llamado diferenciación de imágenes, que consiste en comparar el valor de cada pixel de dos imágenes que tienen un retraso entre ellas, de tal forma que al no existir diferencia se puede concluir que no ha

habido movimiento. En el momento que se presenta una diferencia, se asume que hay un producto dentro del campo de visión de la cámara. En ese momento se inicia el almacenamiento de información. Si se hace el análisis de todos los pixeles de la imagen, el proceso toma más tiempo del conveniente, por esta razón en el proyecto se selecciona un área específica de análisis, esta región es una columna de pixeles ubicada en el centro de cada imagen. El código utilizado para detectar movimiento en el proyecto es el siguiente:

```
movimiento = false;
absdiff(siguiente, actual, d1);
absdiff(actual, previo, d2);
bitwise_xor(d1, d2, resultado_xor);
threshold(resultado_xor,resultado_xor,140,255,CV_THRESH_BINARY);
for(int i = 790; i < 810; i++)
   for(int j = 10; j < 1190; j++)
      if(resultado_xor.at<int>(j,i)>0)
      {
         movimiento = true;
         break;
      }
      else
         mov=0;
```

4.2.3 Localización del código de barras

La localización del código de barras pude realizarse de diversas maneras. La que se escogió para este proyecto tiene como fundamento la unión de las barras para formar una figura sólida donde se encuentra tal código, para así encontrar su contorno y, por consiguiente, sus límites en la imagen.

Suavizado de la imagen

El suavizado es la primera operación que se realiza sobre la imagen donde se encuentra el código de barras, y tiene como objetivo eliminar todo ruido que pueda afectar la localización. Se realiza con el comando GaussianBlur, el cual realiza una media con un Kernel de 3x3 y se usan los parámetros sigma y A en 0, para que OpenCV escoja automáticamente el ancho de la campana. Se usó la siguiente línea de programación donde "origen" es la matriz que contiene la imagen del código de barras y "destino" es la matriz donde se almacena la imagen procesada.

```
GaussianBlur( origen ,destino ,Size(3,3), 0, 0,BORDER_DEFAULT);
```

En la figura 4–1 se puede observar la cara de la caja de un medicamento, en donde está el código de barras, luego de la operación de suavizado.

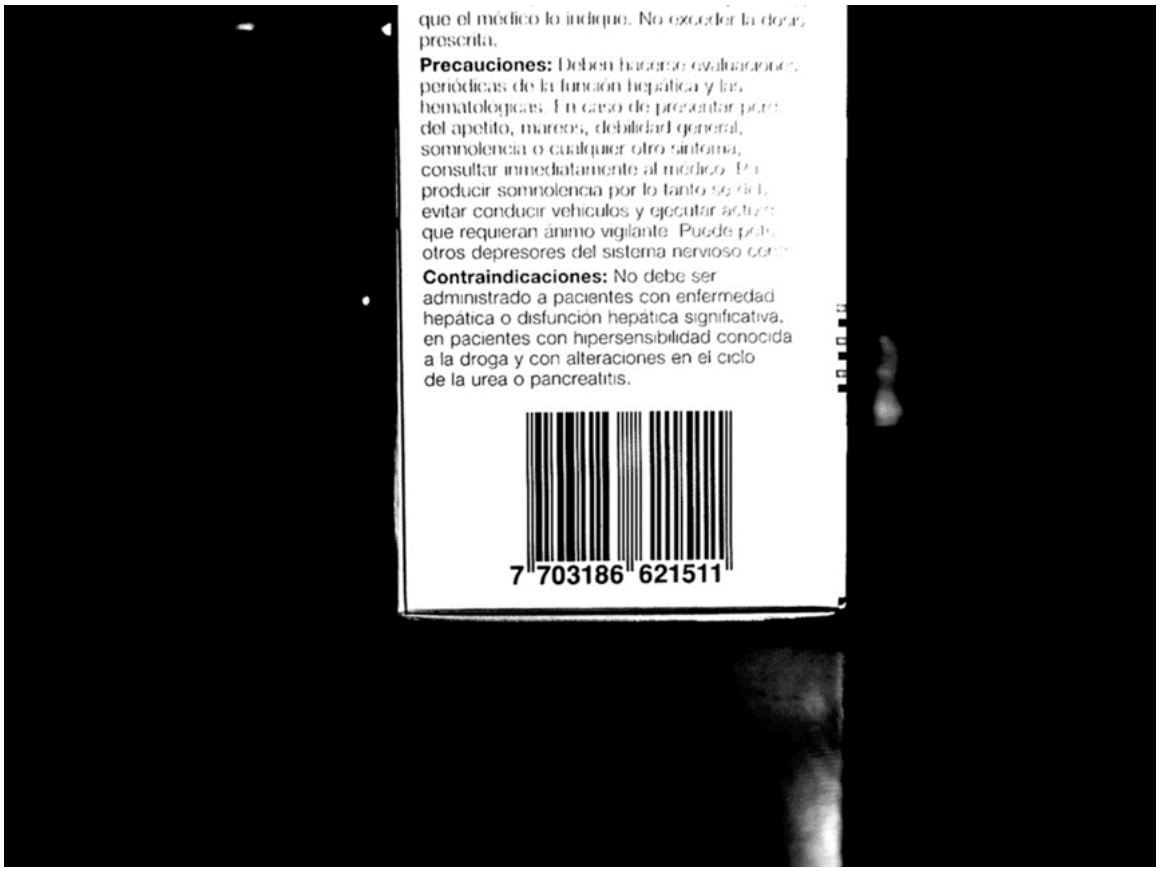

Figure 4–1: Código de barras con suavizado GaussianBlur

4.2.4 Escala de grises

En la Figura 4–1 se ve una imagen que parece estar en escala de grises, ya que está formada de blanco, negro y valores intermedios. Pero la realidad es que es una imagen RGB, en donde cada valor de la matriz, que representa un pixel, contiene 3 componentes (rojo, verde, y azul), que van de 0 a 255, cada uno. La imagen resulta de esta forma porque al escoger los parámetros de la cámara, se definió la saturación en 0, por lo que en cada pixel los 3 componentes siempre tienen el mismo valor, [0,0,0] para negro, [255,255,255] para blanco, y ([00,100,100] para un valor intermedio, por ejemplo. Para poder realizar las operaciones necesarias para localizar el código de barras, es preciso tener la imagen en escala de grises para que solo haya un componente asignado a cada pixel, 0 a negro y 255 a blanco (en imágenes de 8 bits). Opencv cuenta con la función cvtColor que convierte la imagen de un espacio de colores a otro. En este proyecto se usó el siguiente código.

```
cvtColor(imagen, imagen_gris, CV_RGB2GRAY);
```

CV_RGB2GRAY especifica el cambio de RGB a escala de grises, imagen es matriz que se quiere modificar, e imagen gris es la salida de escala de grises de imagen.

4.2.5 Detección de bordes

Los códigos de barras deben tener un alto contraste para detectar fácilmente los ceros y los unos y así poder ser leídos. Esta característica permite eliminar información de las imágenes donde no haya un alto contraste. Un filtro Sobel es aplicado a la imagen en escala de grises, para obtener los bordes de la imagen. Ver figura 4–2. Se eliminan las áreas de bajo contraste, pero no se pueden eliminar las letras ni los bordes de la caja.

4.2.6 Threshold

El Threshold se aplica por 2 razones. La primera es lograr quitar áreas de en donde el operador Sobel devolvía zonas con intensidad baja, porque había un cambio

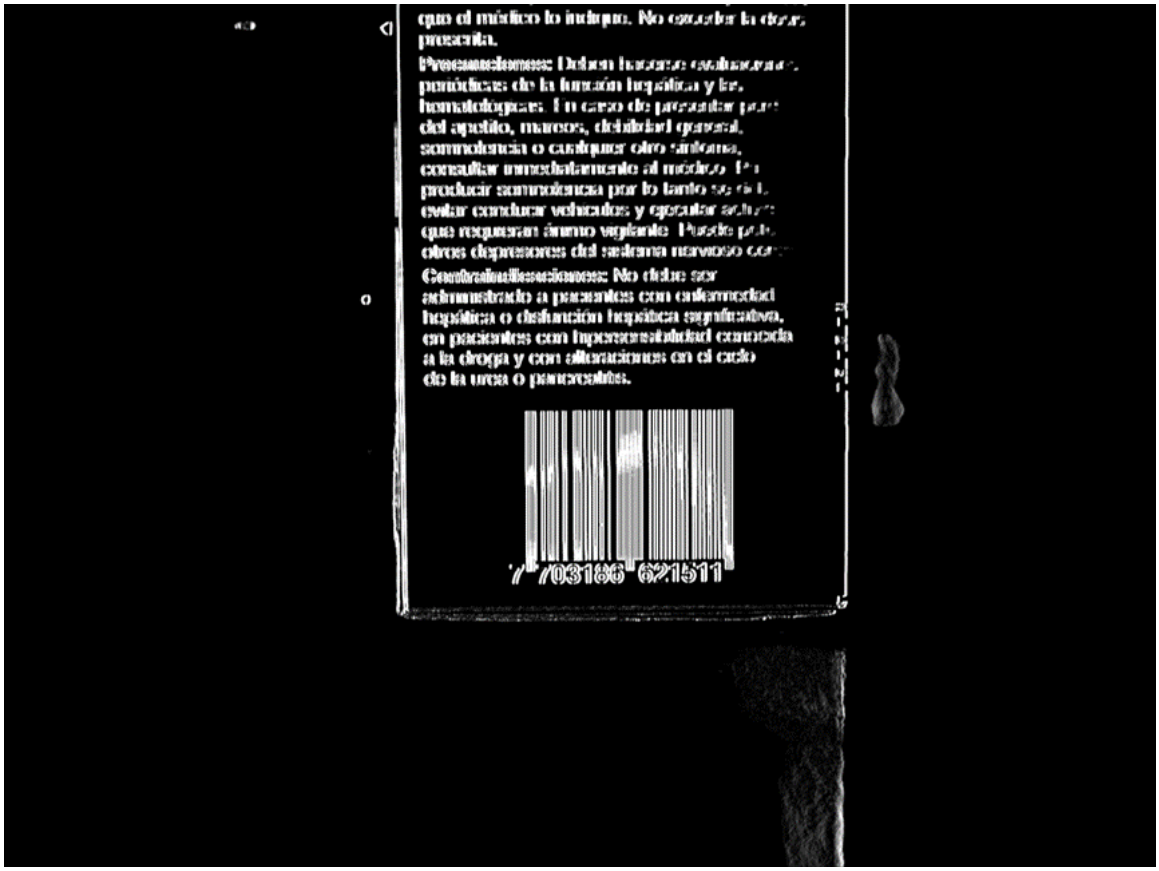

Figure 4–2: Detección de bordes con Sobel

no tan abrupto de intensidad. La segunda razón es obtener una imagen binaria, es decir con solo valores de blanco y negro. Esto porque solo se necesitan estos dos valores para conocer la ubicación del código de barras, y poder utilizar las operaciones morfológicas fácilmente. Se usa el comando threshold en la imagen después de haber obtenido el gradiente con Sobel y se define un umbral de 100 se obtiene la imagen de la figura 4–3.

```
threshold(gradiente_sobel,imagenbinaria,100,255,CV_THRESH_BINARY);
```

El parámetro gradiente_sobel es la matriz que contiene el resultado de la operación Sobel, imagen binaria es donde se almacena el resultado, el umbral es 100, es decir que los pixeles con intensidad menor a 100 se vuelven negros. 255 es el valor máximo que se asignan a los valores mayores a 100 y CV_THRESH_BINARY es el tipo de operación que hace.

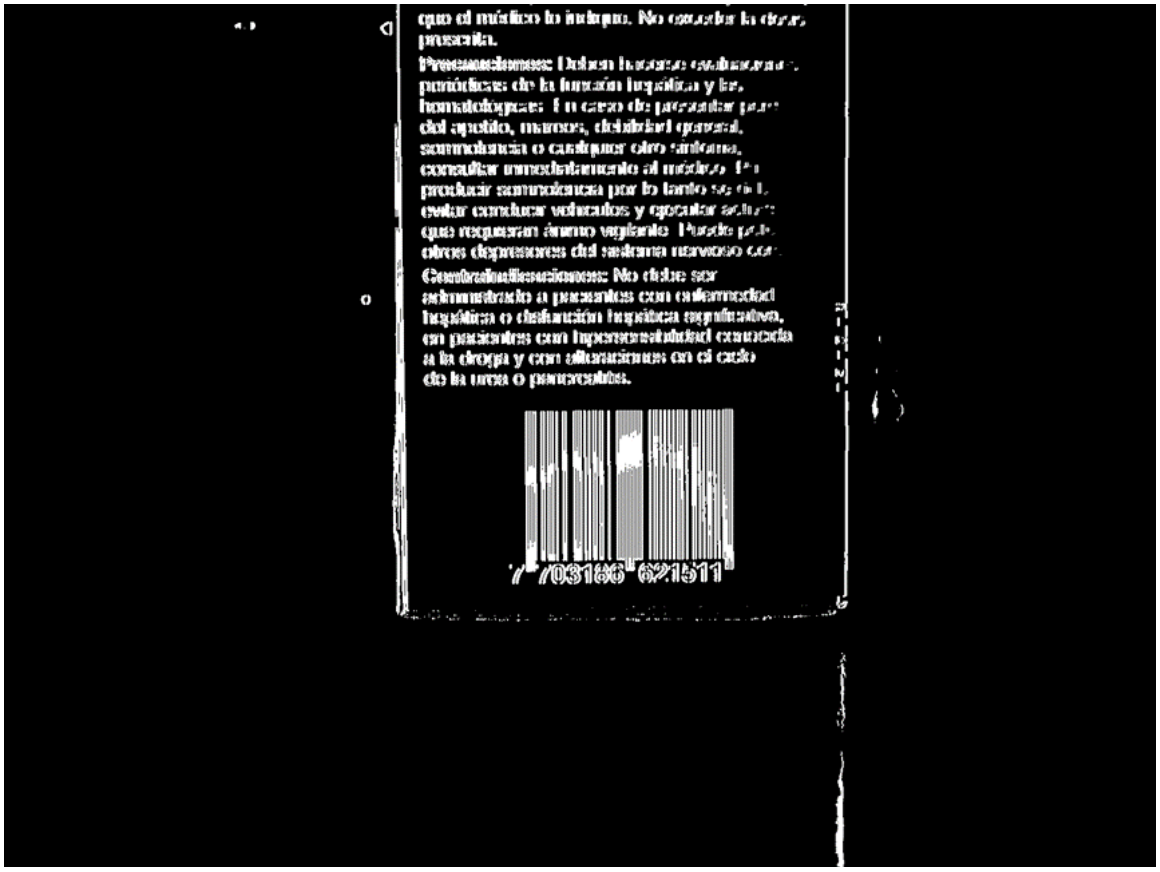

Figure 4–3: Threshold a los bordes de la caja

4.2.7 Operaciones morfológicas

Para ubicar el código de barras utilizando operaciones morfológicas, hay que lograr unir las barras para formar un elemento solido en donde se encuentre el código de barras. Esto se logra con la operación dilatación. Esta operación necesita un elemento estructurado para poder llevarse a cabo. El tamaño del elemento se debe escoger de tal forma que pueda unir las dos barras más separadas de cualquier código de barras. Tras varias pruebas se concluyó que a la distancia de operación, la mayor distancia entre dos barras es de 19 pixeles, por esta razón se escoge un tamaño de 19x19.

```
elemento = getStructuringElement(MORPH_RECT, Size(19,19), Point(9,9));
```

El parámetro MORPH_RECT quiere decir que el elemento estructurado va a tener forma rectangular (bien pudo haberse escogido circular), y el punto ancla va a ser el centro del elemento estructurado.

Después de haber definido este elemento es posible realizar la operación de dilatación de la siguiente forma:

```
dilate(imagenbinaria, resultado, elemento);
```

Esta operación toma la imagen binaria, se hace la dilatación con el elemento seleccionado y la guarda en resultado (figura 4–4).

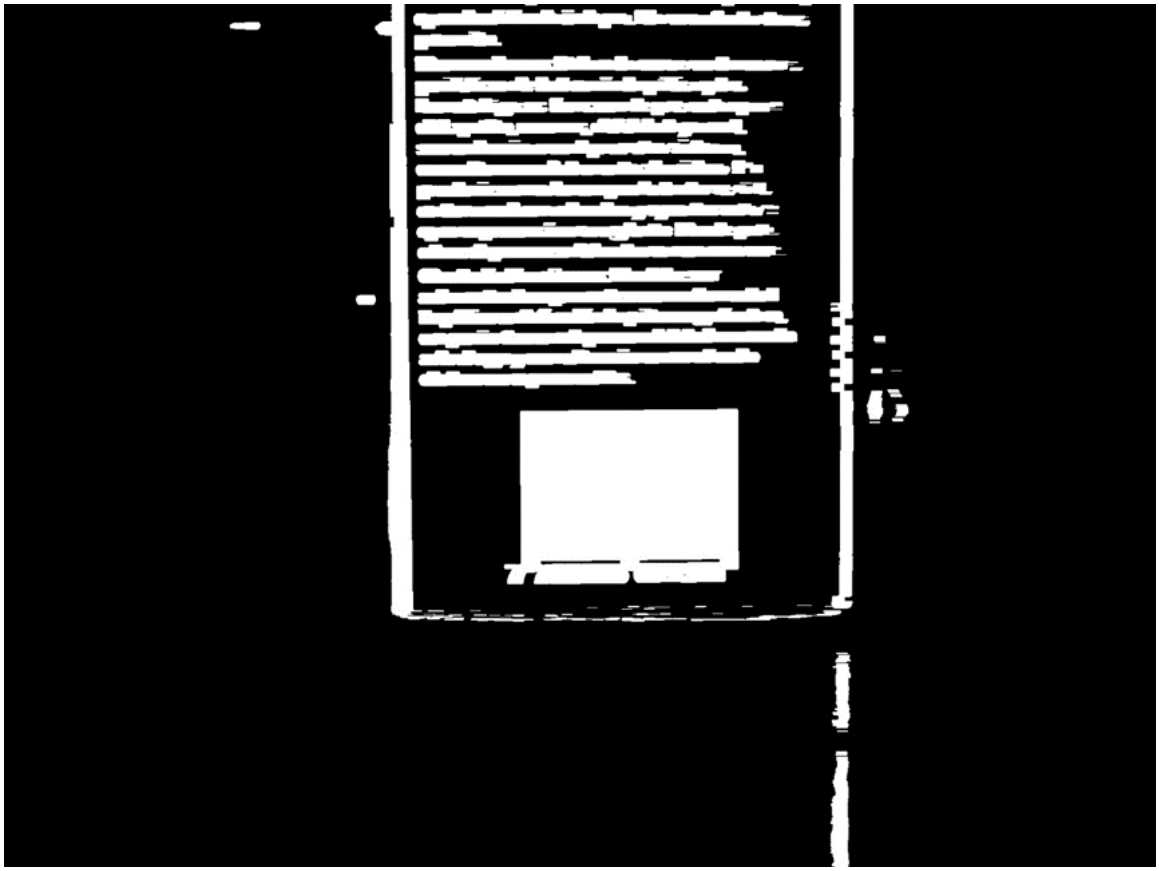

Figure 4–4: Dilatación del Threshold

Para eliminar la información adicional al código de barras se aplica la operación de *erosion*, pero el tamaño del elemento tiene que asegurar que la erosión sea lo suficientemente grande para eliminar las letras y demás, pero no demasiado grande para eliminar el código de barras. Con el siguiente código se obtiene la figura 4–5.

```
elemento = getStructuringElement(MORPH_RECT,Size(81,147),Point(40,73));
erode( resultado, resultado, elemento );
```

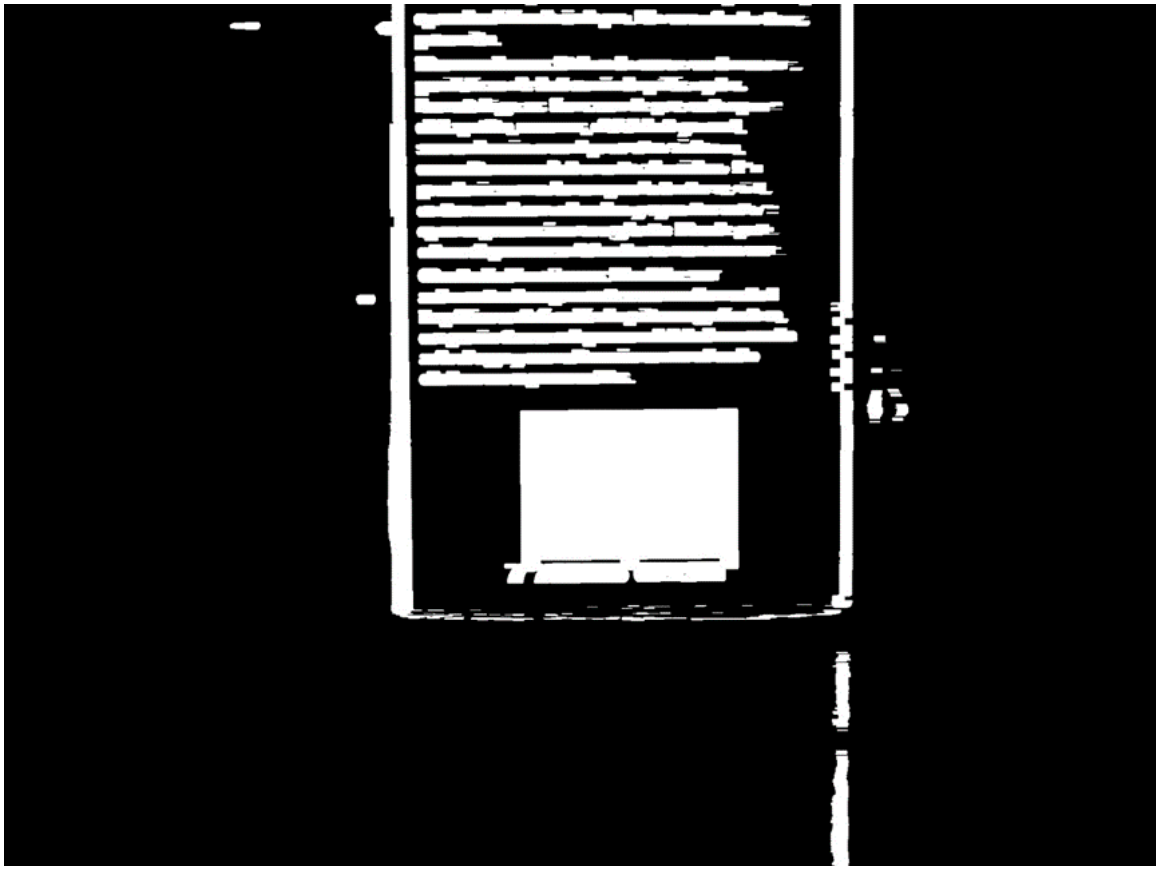

Figure 4–5: Eliminar elementos por Erosión

Por último, nuevamente se hace una operación de dilatación, para volver a obtener el tamaño completo del código. Como se ve en la figura 4–6

```
elemento = getStructuringElement(MORPH_RECT,Size(101,161),Point(50,80));
dilate( resultado, resultado, elemento);
```

4.2.8 Encontrar contornos

Luego de obtenerse la región donde se encuentra el código de barras es necesario recortarlo para enviarlo a ZBar. Se aplica findContours para determinar el contorno de la región con el siguiente código:

```
findContours(resultado,contournos,CV_RETR_EXTERNAL,CV_CHAIN_APPROX_NONE);
```

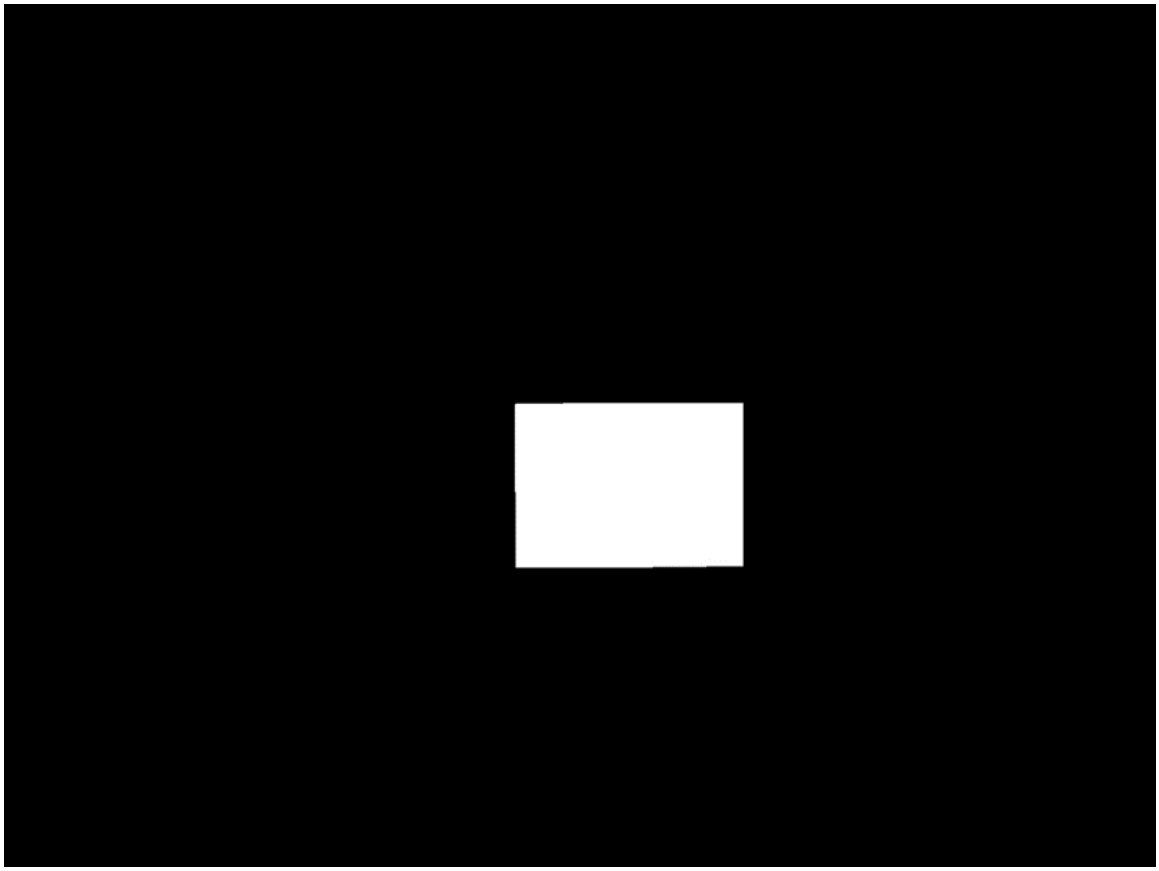

Figure 4–6: Selección de ubicación del código de barras

Se toma la imagen "resultado" y se guarda un vector de puntos llamado "contornos". CV_RETR_EXTERNAL, devuelve los contornos externos, y CV_CHAIN_APPROX_NONE, almacena en "contornos" todos los puntos del contorno. Una ilustración de cómo funciona findContours se hace con la figura 4–7.

4.2.9 Comunciación con Zbar

Una vez que se tiene recortada exclusivamente la región que contiene el código de barras, se hace la decodificación de la simbología por medio de un software dedicado a cumplir esta función, en el proyecto se escogió ZBar como programa lector de código, ya que es de código abierto. Tiene excelente velocidad de respuesta, reconoce las simbologías más populares en la industria, es compatible con varias arquitecturas como Windows o Linux y utiliza el digito de verificación para comprobar la lectura. Para poder decodificar la información, es necesario que ZBar reciba la imagen desde

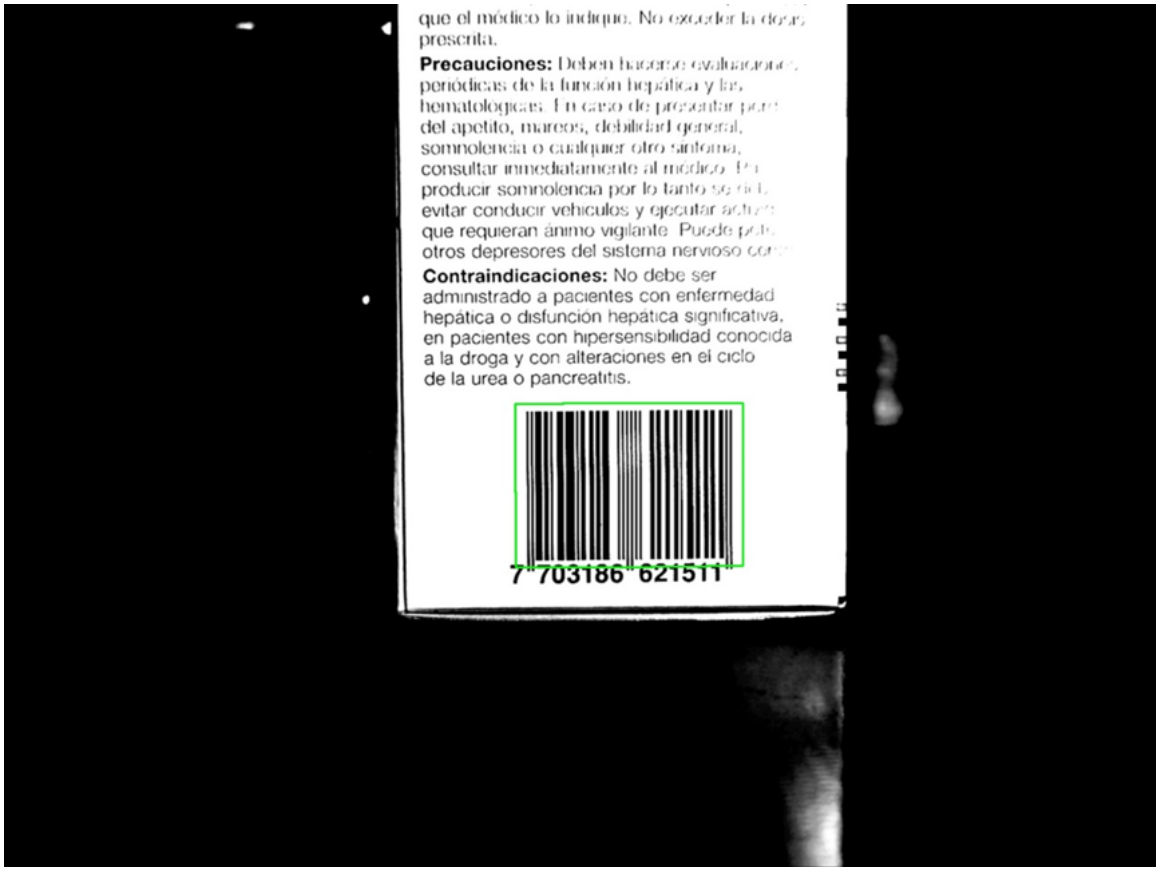

Figure 4–7: Demostración de la ubicación del código de barras

el código principal. Esto se hace creando una comunicación entre los dos programas que se denomina "pipe". Al crear esta conexión se pueden enviar y recibir datos en ambos sentidos, de esta forma se puede enviar la imagen recortada al decodificador y él puede enviar la simbología extraída como una cadena de caracteres. Las líneas de código utilizadas para cumplir esta función fueron las siguientes:

```
QString Dialog::exec(char* cmd) {
    FILE* pipe = popen(cmd, "r");
    if (!pipe) return "ERROR";
    char buffer[128];
    QString result = "";
    while(!feof(pipe)) {
        if(fgets(buffer, 128, pipe) != NULL)
            result += buffer;
    }
```

```
    pclose(pipe);
    return result;
}

respuestazbar = exec(nombre_archivo2);
```

4.2.10 Recolección de datos

En el momento que se detecta movimiento con el código descrito anteriormente, se habilita el almacenamiento de las imágenes, en este punto es posible que se tomen varias fotos del mismo producto. Esto conlleva a dos resultados, primero se genera la posibilidad de comparar el resultado de varias imágenes y corroborar que la simbología decodificada es la correcta. Pero también se puden obtener varios códigos del mismo producto, lo cual puede crear confusión en el sistema, ya que a pesar de que solo se esté analizando un medicamento se obtienen varias respuestas que podrían indicar la presencia de varios productos del mismo tipo. Para gestionar eficientemente este comportamiento, se toman medidas que aseguran un conteo adecuado del pedido. Para evitar que las múltiples simbologías leídas de un mismo producto se tomen como múltiples productos del mismo tipo, se condiciona la detección del medicamento, se genera una variable booleana que cambia de estado cuando se detecta movimiento, y solo regresa a su estado inicial cuando deja de detectar la presencia del producto, de esta forma mientras que esa variable no esté en su estado de reposo no se concluye que el mismo producto ha salido del campo de visión de la cámara. Conociendo cuándo se cambia de producto, se pueden aprovechar todas las respuestas que genere ZBar de este ítem, comparándolas entre ellas y utilizando la moda de estos datos, como la información verídica de la lectura, con este proceso se eliminan las lecturas erróneas que se presenten en la decodificación.

4.2.11 Comunicación con la base de datos

El primer paso para iniciar el proceso de verificación consiste en comunicarse con la base de datos en donde se encuentran las órdenes de despacho, y seguidamente se obtienen los datos del pedido que se va a verificar. Esta información se almacena en el programa desarrollado en el proyecto, con el fin de tener la posibilidad de comparar los productos a los que se les está haciendo la lectura del código de barras con los medicamentos que se encuentran presentes en la lista solicitada. El sistema inicia así con el fin de hacer la comparación de cada medicamento por separado, de tal forma que en el caso de presentarse un medicamento que no esté en el listado se pueda generar una señal de alerta de forma inmediata. Igual ocurre cuando la cantidad de un medicamento específico supera el valor del producto solicitado. En los demás casos, el resultado de la comparación se imprimirá en pantalla cunado la orden se verifique en su totalidad y el operario indique al sistema que se finalice el proceso de verificación El código que genera la comunicación con la base de datos es:

```
QSqlDatabase db = QSqlDatabase::addDatabase("QMYSQL");
        db.setHostName("192.168.1.13");
        db.setDatabaseName("tesisdb");
        db.setUserName("tesis");
        db.setPassword("t3s1s");
        if(!db.open())
        {ui->textBrowser->append("error");}
        else
        {ui->textBrowser->append("hola");}

        QSqlQuery query;
        //query.result();
        //query.prepare("SELECT * FROM codbarras");
        if(!query.exec("SELECT * FROM codbarras"))
        {ui->textBrowser->append("consulta error");}
```

```
        else{
        while (query.next()) {
            QString numero = query.value(0).toString();
            //doSomething(country);
            ui->textBrowser->append(numero);
        }}
}
```

Para lograr la comunicación con el servidor de Offimedicas S.A. se utilizó el sistema de gestión de base de datos MySQL, ya que la información de la compañía es manejada con este mismo programa.

4.2.12 Verificación de resultados

La verificación del despacho se hace comparando dos listas de productos. La primera es la que se extrae de la base de datos en donde se encuentran los medicamentos que ya están facturados. La segunda es la lista que se genera en el sistema desarrollado.

4.2.13 Verificación de resultados

Se creó la interfaz gráfica de tal forma que los usuarios de este sistema alcanzarán el mayor grado de confort y de comodidad al momento de hacer uso del mismo. Pensando en los usuarios, se requiere que sea de fácil uso, que permita una interacción amigable, y sobre todo que dé los resultados para los cuales fue creado este sistema. Solo se requiere un botón para iniciar todo el proceso, además tiene una imagen que facilita la verificación visual de la respuesta del sistema, la cual ayuda al operario a excluir, si es necesario, algún medicamento. La GUI tiene un editor de texto que se usa para ingresar el número del pedido que se va a analizar, una vez registrada la orden se oprime el botón de inicio y comienza la verificación. Con cada medicamento se muestra el código leído en una ventana de impresión. (ver figura 4–8)

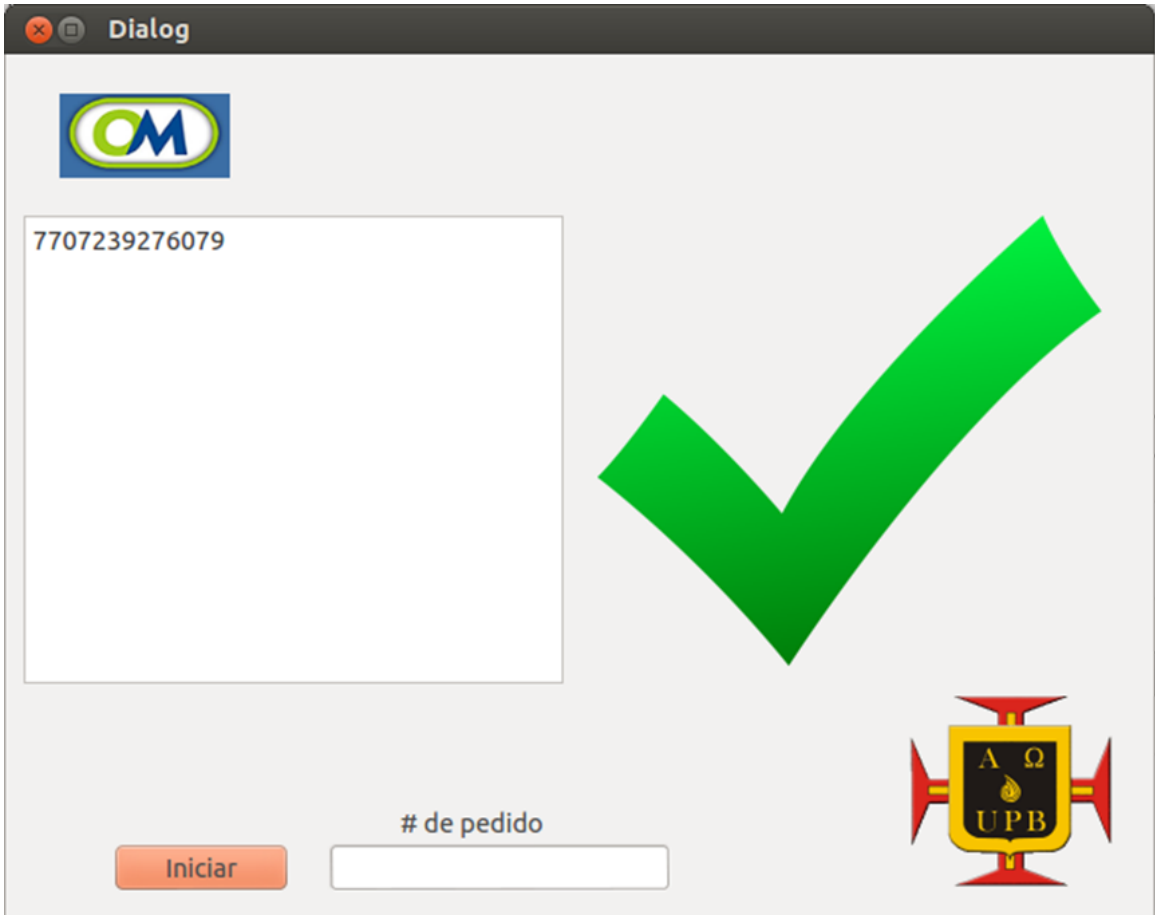

Figure 4–8: GUI del sistema

CAPITULO 5
PRUEBAS

La preocupación principal al iniciar el proyecto fue la lectura de los códigos de barras, por esto se inició la investigación buscando formas para cumplir esta función y como resultado se seleccionó ZBar como método para decodificar la información. Luego de verificar que ZBar era capaz de satisfacer la necesidad de leer las simbologías utilizadas en el proceso, había que comprobar la velocidad máxima a la cual puede leer el código de barras correctamente. Para conocer la velocidad óptima de funcionamiento se tomaron fotos de códigos de barras en movimiento, utilizando una cámara digital y una banda transportadora. Inicialmente, se definió una velocidad del motor que movía la banda de 3Hz, si la imagen era lo suficientemente nítida se aumentaría la velocidad, si no se disminuiría. Los resultados de este proceso no fueron satisfactorios, ya que la imagen seguía saliendo corrida (efecto de trepidación) a la velocidad mínima del motor y no era apta para la lectura del código de barras. Ver figura 5–1

De la investigación realizada acerca de fotografía en movimiento, se concluyó que era necesario tener en cuenta parámetros de configuración de la cámara, los cuales mejoran las capturas en movimiento. El principal parámetro que se debe modificar es el tiempo de exposición, el cual controla el tiempo que el obturador permite que la luz entre en la cámara, lo cual permite obtener fotos nítidas en movimiento. Al aumentar la velocidad de obturación también es necesario aumentar la iluminación, para que la imagen pueda quedar clara. Luego de definir estos parámetros, había

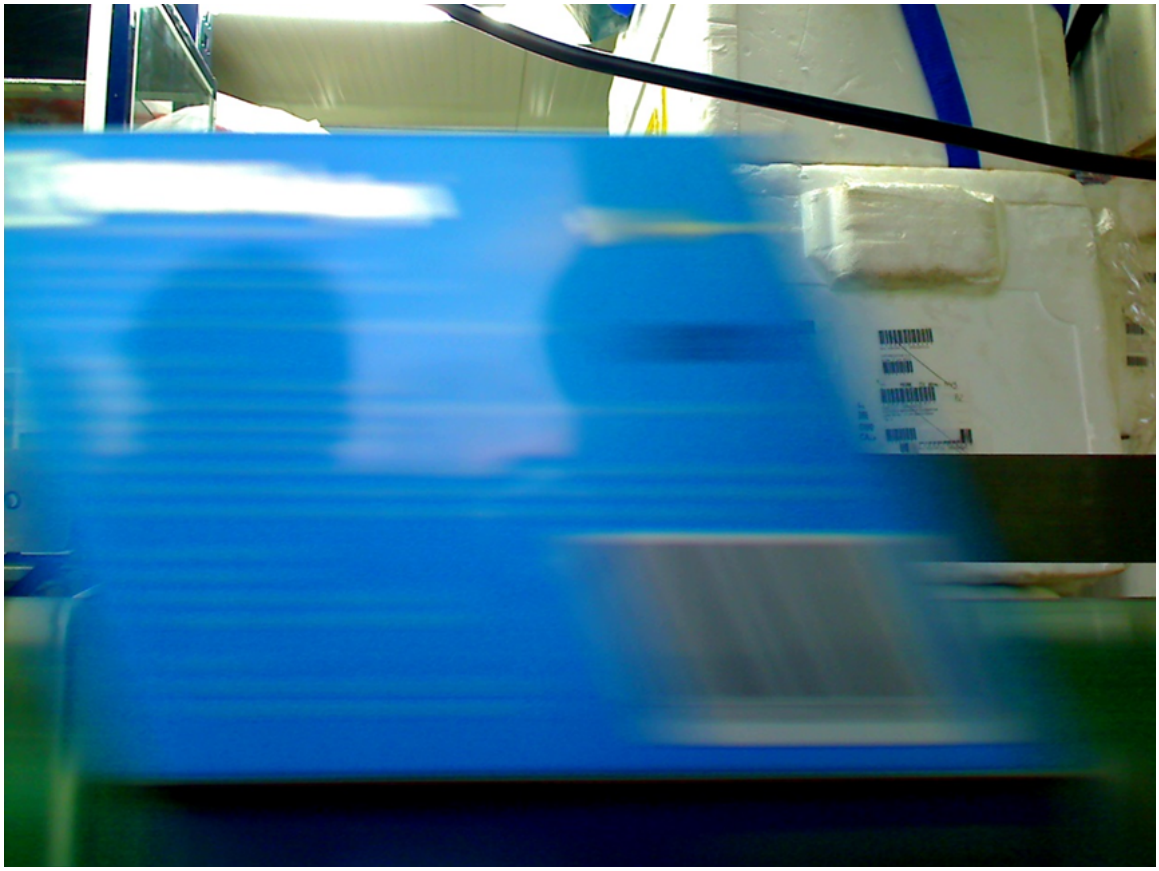

Figure 5–1: Efecto de trepidación

que establecer otras variables, principalmente la distancia focal y resolución, para así poder definir la velocidad de la banda. La resolución de las pruebas iniciales fue de 8 megapíxeles (ver figura 5–2), las imágenes obtenidas permitían la lectura de la simbología, pero el tiempo requerido para la lectura era demasiado largo (2.6 segundos. Ver figura 5–3). Con un tiempo de 2.6 segundos no es posible garantizar la captura de todos los productos sobre la banda transportadora, ya que mientras se procesa una imagen de este tamaño algún medicamento podría pasar por el campo de visión de la cámara sin ser detectado.

Para disminuir el tiempo de procesamiento se redujo la resolución hasta 640X480 pixeles. Con este tamaño la imagen no tiene la suficiente nitidez para permitir la lectura de códigos con un factor de magnificación menores a uno. Luego de varias pruebas se concluyó que la menor resolución viable para el desarrollo del proyecto es 2

Figure 5–2: Foto de 8 megapíxeles

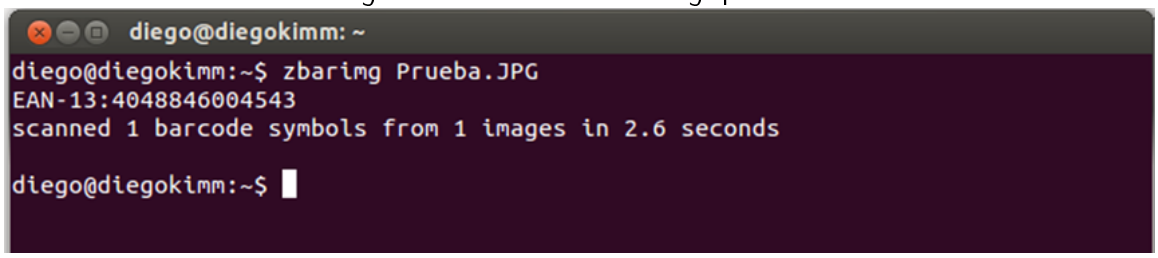

Figure 5–3: Duración de ZBar para leer el código de barras en la foto de 8 megapíxeles

megapíxeles, ya que permite la decodificación para cualquier factor de magnificación (con una distancia focal fija) y acorta el tiempo de procesamiento. (ver figuras 5–4 y 5–5)

Para definir la distancia entre los códigos de barras y la cámara se hicieron pruebas de funcionamiento y se escogió 9 centímetros como medida de esta variable, ya que con este valor es posible leer los códigos más pequeños y más grandes utilizados en las pruebas. A esta distancia y con el tamaño determinado anteriormente, se definió que la velocidad máxima de los productos en la banda es de 15 centímetros

Figure 5–4: Foto de 2 megapíxeles

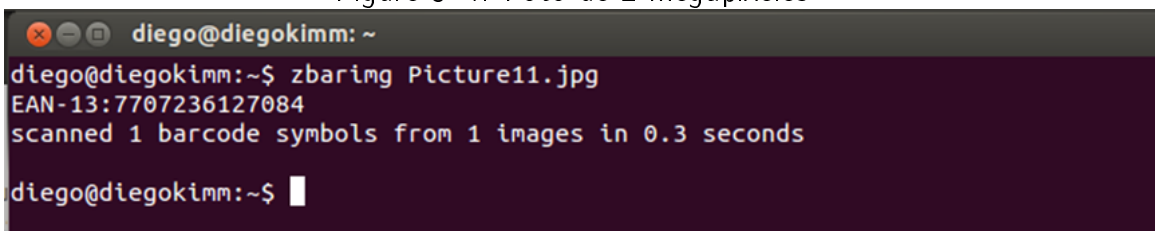

Figure 5–5: Duración de ZBar para leer el código de barras en la foto de 2 megapíxeles

por segundo. A pesar de disminuir sustancialmente el tiempo de procesamiento, no es suficientemente rápido para esta aplicación. Por esta razón la investigación se enfocó en buscar un método para reducir este tiempo. De la investigación se encontraron varias publicaciones relacionadas con la localización del código de barras[5][6][7][8]. En los artículos había diversos métodos para esta función.

De los artículos estudiados y basados en las pruebas realizadas por[8] se concluye que la forma más eficiente de localizar el código de barras es usando operaciones morfológicas, por esta razón, se desarrolló el procesamiento basado en este método.

Fue necesario probar constantemente los resultados de estas operaciones morfológicas, ya que de éstos depende de la modificación de variables que afectan a dichas funciones. Para valores de los elementos estructurados no adecuados, es posible obtener contornos que no contienen el código de barras (figura 5–6). Con la configuración apropiada se evita este problema.

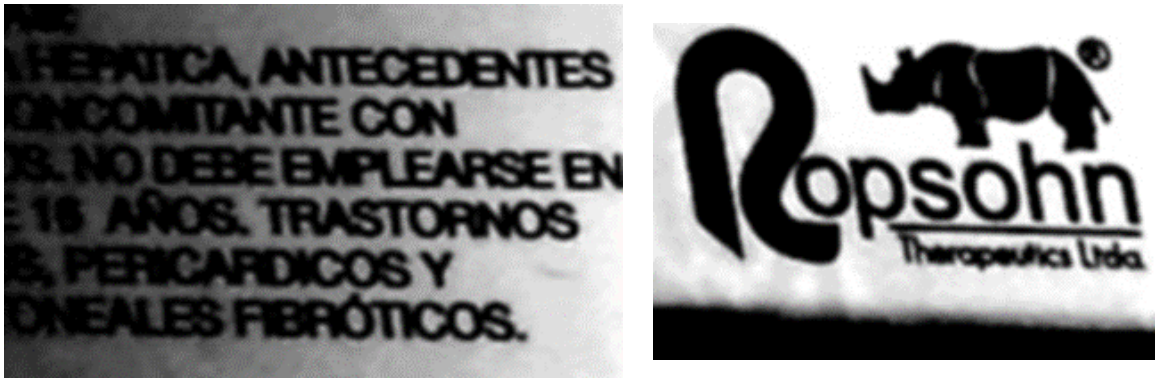

Figure 5–6: Falsa deteccion de contornos de códigos de barras

La imagen que contiene exclusivamente el código de barras debe ser procesada para obtener la simbología. Esta tarea la realiza ZBar, por esta razón se debe encontrar una manera de enviarle la imagen, para que regrese la respuesta de la lectura. La primera opción probada fue Shell scripting, con la cual se usan comandos del terminal de Linux para ejecutar aplicaciones y hacer tareas de programación. La idea era hacer un bucle, que transfiriera la información entre el software de detección del código de barras y ZBar. Esta opción se descartó puesto que cada vez que se ejecutaba el software desarrollado, la cámara requería mucho tiempo para su inicialización. Para solucionar este problema, su buscó otra alternativa, y se escogió crear una comunicación entre los software utilizando un pipe. Las tuberías o pipes simplemente conectan la salida de un proceso con la entrada de otro. De esta forma se le entrega la imagen recortada a Zbar y la simbología retorna al sistema. Con la imagen procesada del código de barras, el tiempo empleado por Zbar disminuyó un 90% (Ver figura 5–9) entre la imagen de la figura 5–7 y la imagen del código recortado (figura 5–8).

Figure 5–7: Fotografia completa

Figure 5–8: Código recortado

```
diego@diegokimm:~$ zbarimg imagen_156.jpg
EAN-13:7704588000010
scanned 1 barcode symbols from 1 images in 0.36 seconds

diego@diegokimm:~$ zbarimg recortada_156.jpg
EAN-13:7704588000010
scanned 1 barcode symbols from 1 images in 0.04 seconds

diego@diegokimm:~$
```

Figure 5–9: Tiempo de lectura de la imagen completa y el código recortado

CAPITULO 6
RESULTADOS

6.1 PLANO DEL SISTEMA DE VERIFICACIÓN

Se creó un diseño tridimensional del prototipo construido, con el fin de ilustrar el funcionamiento del sistema. (ver figura6–1 y 6–2)

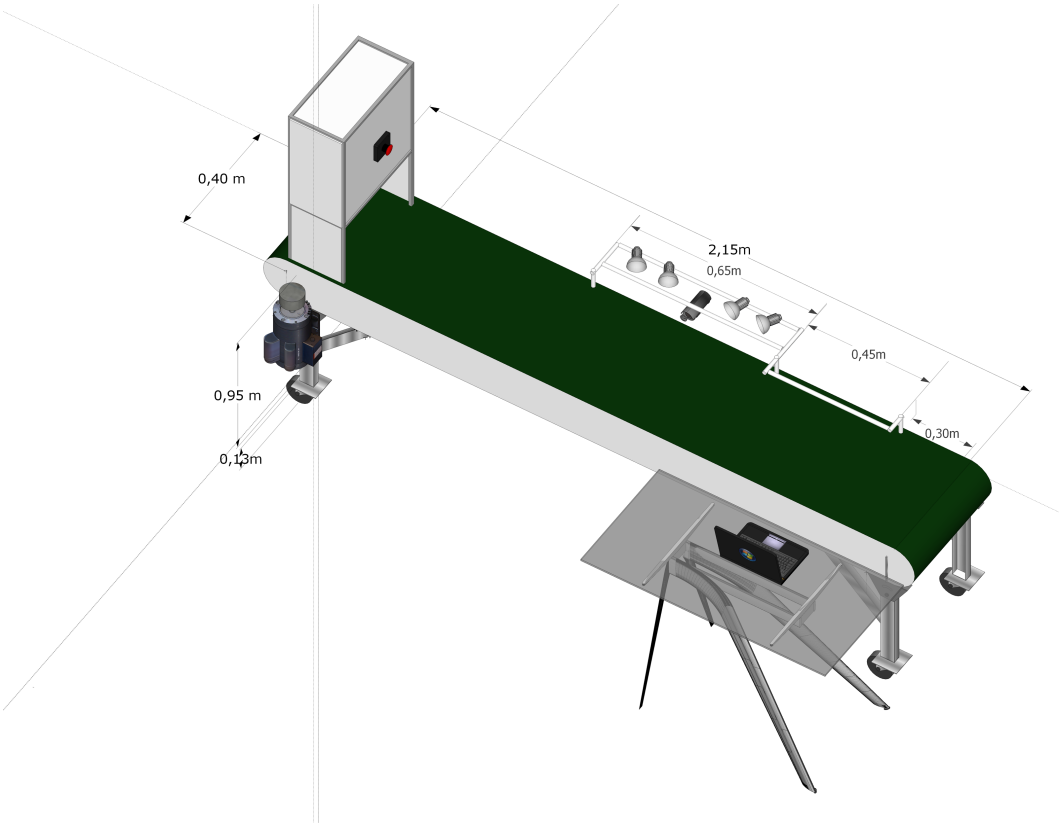

Figure 6–1: Plano de Sistema

6.2 DEMOSTRACIÓN

Las pruebas de funcionamiento dieron como resultado una lectura del 100% de los medicamentos que vienen empacados en cajas. Se realizaron múltiples pruebas

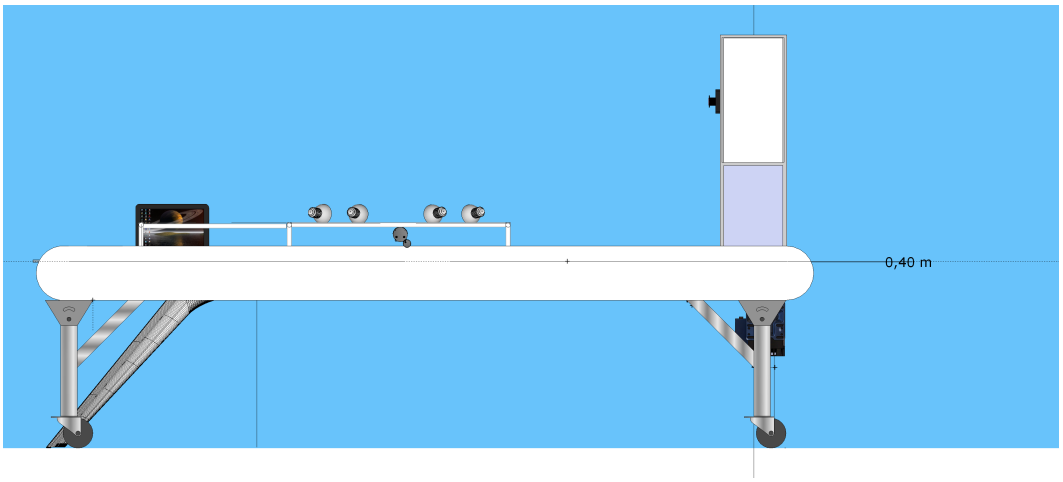

Figure 6–2: Plano de Sistema

con diferentes productos que se comercializan en cajas. En la figura 6–3 se observan ejemplos de la localización del código en una imagen compleja.

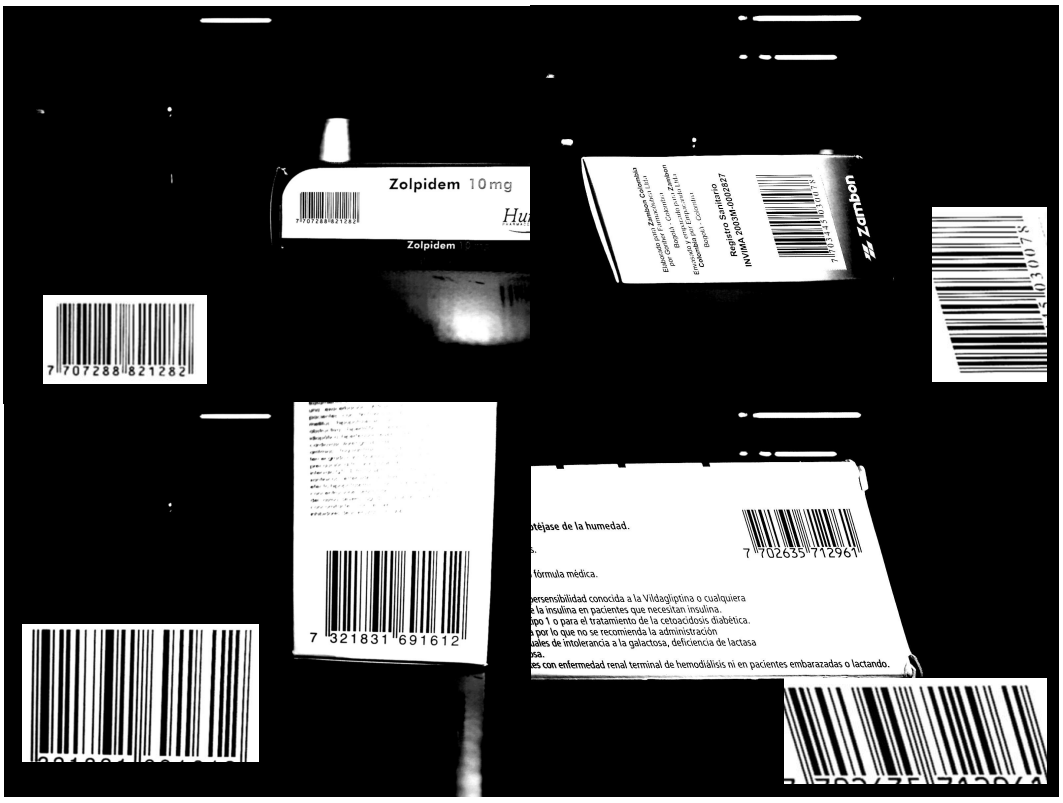

Figure 6–3: Demostración de localización de los códigos

Para envases cilndricos, todava se obtienen casos en los cuales el código de barras no se logra ubicar, y otros en donde, a pesar de ser localizados, el resultado de la lectura es infructuoso. Esto se debe a que la curvatura del envase donde est

impreso el código impide la distribución de luz sobre la superficie uniformemente. En la figura 6–4 hay un ejemplo de un código de barras localizado, pero no ledo, y otro donde no fue posible ubicar el código.

Figure 6–4: Demostración de localización de los códigos en frascos

CAPITULO 7
CONCLUSIONES Y TRABAJOS FUTUROS

Teniendo en cuenta los objetivos propuestos para este proyecto, se afirma que se dio cumplimiento a la totalidad de los mismos. Se seleccionaron los dispositivos adecuados para la aplicación desarrollada, ya que los resultados cumplieron con las metas de desempeño planteadas. Se construyó un prototipo funcional del sistema de verificación capaz de cumplir todas las tareas propuestas. Se hizo un plano en tercera dimensión del prototipo. Se generó un enlace de comunicación entre la base de datos de Offimedicas S.A. y el sistema de verificación y se realizaron pruebas del sistema, utilizando los elementos seleccionados.

El tamaño de la imagen que se analiza define el tiempo empleado en su procesamiento, por esta razón se redujo la resolución, teniendo en cuenta el límite de nitidez para la decodificación.

Se encuentran productos con cdigos de barras del sistema EAN13, que no cumplen con las normas del factor de magnificación, ya que se trabajó con códigos más pequeños que los reglamentarios, por lo que fue necesario adecuar la distancia focal del sistema, y el valor del elemento estructurado que realiza la erosión.

Se planea incorporar la lectura de medicamentos con envases cilíndricos, mejorando las condiciones de luminosidad y el dispositivo usado para captar los códigos.

Se implementarán más cámaras al sistema para que la decodificación se pueda hacer sin necesidad de dar una orientación específica al producto.

Con el procesamiento de imágenes se analizarán propiedades de los productos como estado físico y fechas de caducidad.

REFERENCIAS BIBLIOGRÁFICAS

[1] Juan Pablo AntÚn. *Logística de distribución física a minoristas*. UNAM, 2005.

[2] Julio Juan Anaya Tejero. *Logística integral: la gestión operativa de la empresa*. ESIC editorial, 2007.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'reilly, 2008.

[4] Robert Laganière. *OpenCV 2 computer vision application programming cookbook*. Packt Publishing, 2011.

[5] Pavel Šimurda. Barcode localization in image. In *Proceedings of the 17th Conference STUDENT EEICT 2011 Volume*, volume 1, pages 169–171, 1111.

[6] Ruben Muniz, Luis Junco, and Adolfo Otero. A robust software barcode reader using the hough transform. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pages 313–319. IEEE, 1999.

[7] Chunhui Zhang, Jian Wang, Shi Han, Mo Yi, and Zhengyou Zhang. Automatic real-time barcode localization in complex scenes. In *Image Processing, 2006 IEEE International Conference on*, pages 497–500. IEEE, 2006.

[8] Melinda Katona and László G Nyúl. A novel method for accurate and efficient barcode detection with morphological operations. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pages 307–314. IEEE, 2012.