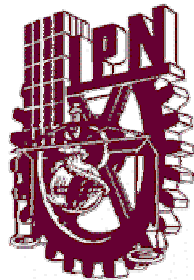


INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN



**DISEÑO DE UN SISTEMA DE CAPTURA Y
PREPROCESAMIENTO DE IMÁGENES
DIGITALES**

T E S I S

**QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS EN INGENIERÍA DE CÓMPUTO
CON ESPECIALIDAD EN SISTEMAS DIGITALES**

**P R E S E N T A
ISRRAEL MOLINA ESPINOSA**

**DIRECTOR DE TESIS
AMADEO JOSÉ ARGÜELLES CRUZ**

México D. F. 2003



INSTITUTO POLITECNICO NACIONAL

COORDINACION GENERAL DE POSGRADO E INVESTIGACION

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 17 del mes de Diciembre del 2002 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"DISEÑO DE UN SISTEMA DE CAPTURA Y PREPROCESAMIENTO DE IMÁGENES DIGITALES"

Presentada por el alumno:

MOLINA

Apellido paterno

ESPINOSA

materno

ISRRAEL

nombre(s)

Con registro:

A	9	9	0	2	5	1
---	---	---	---	---	---	---

aspirante al grado de: **MAESTRO EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON ESPECIALIDAD EN SISTEMAS DIGITALES**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACIÓN DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente



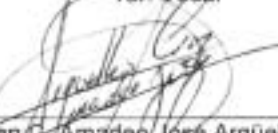
Dr. Cornelia Yáñez Márquez

Secretario



M. en C. Oscar Camacho Nieto

1er. Vocal



M. en C. Amadeo José Argüelles Cruz

2do. Vocal



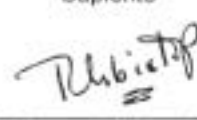
M. en C. Osvaldo Espinosa Sosa

3er. Vocal




M. en C. Pablo Manrique Ramirez

Suplente



M. en C. Romeo Urbieta Parrazales

EL PRESIDENTE DEL COLEGIO



Dr. Juan Luis Díaz de León Santiago
INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION

*Para mi mamá,
Maria de la Luz Espinosa Rodríguez
que me apoyó y permitió
realizar un paso más en mi vida.*

Para mi familia.

AGRADECIMIENTOS

Este trabajo fue posible gracias al apoyo y cooperación de diversas personas, empresas y organizaciones que proporcionaron ayuda, material y los recursos necesarios para llevar a término el desarrollo y construcción del sistema diseñado para esta tesis, en especial quiero agradecer a:

Samsung Semiconductors y su distribuidor ADELSA, por proporcionar los decodificadores de video, componentes principales del sistema.

Texas Instruments, por el desarrollo de los DSP's y del software de diseño utilizados en la implementación de los algoritmos y por proporcionar componentes para la etapa de polarización y regulación de los circuitos.

National Semiconductors, por proporcionar material utilizado en el desarrollo de los sistemas.

XILINX, por su aporte en el área de la lógica programable y de sus programas y dispositivos que permitieron reducir y facilitar el diseño del sistema. Gracias a esto, se logro un tamaño reducido y la posibilidad de modificar la lógica sin adicionar o quitar componentes.

Centro de Investigación en Computación, a los maestros del área de hardware y en especial a todos los compañeros y amigos en el Laboratorio de Equipos Ligeros por toda su ayuda y amistad.

Todos los nombres propios de componentes electrónicos, programas, sistemas operativos, equipo de hardware y documentos que aparecen en este trabajo son marcas registradas de sus respectivas compañías u organizaciones.

RESUMEN

DISEÑO DE UN SISTEMA DE CAPTURA Y PREPROCESAMIENTO DE IMÁGENES DIGITALES

La presente tesis se refiere al concepto de captura y preprocesamiento de imágenes digitales, así como al sondeo de sistemas similares que se encuentran disponibles en el momento del desarrollo del presente trabajo. Esta información sirvió para dar paso al diseño de un sistema constituido de tres etapas: la captura, el preprocesamiento y la visualización de imágenes digitales y de sus resultados. Al explorar el ámbito de cada parte del diseño se procedió a seleccionar para cada caso una de diferentes soluciones posibles. Para el caso de la captura se diseñó y construyó un circuito que captura imágenes a partir de la decodificación de señales de video compuesto; para el caso del preprocesamiento de imágenes se utilizó un sistema de desarrollo basado en un procesador de señales digital o DSP donde se implementaron varios algoritmos que se diseñaron para realizar funciones de preprocesamiento; adicionalmente se implementaron mediante software para la depuración de los propios algoritmos. Finalmente, la visualización de las imágenes se realiza mediante un programa de software hecho a la medida que también se encarga de interactuar con los otros dos sistemas y administrar las imágenes mediante el formato de archivos.

ABSTRACT

DESIGN OF A SYSTEM FOR CAPTURE AND PREPROCESSING OF DIGITAL IMAGES

This thesis refers to capture and preprocessing images concepts, as well as of scanning of similar systems available at the moment of research and development of this job. All the information was used to open the way for the design of a new system which is divided into three main parts: capture, preprocessing and visualization of digital images. Exploring each part of the system leads to select one between several options. For the case of capturing, a circuit was designed and manufactured, taking into account composite video signal decoding; a development board based on DSP was used for images manipulation, where several algorithms were downloaded to generate several preprocessing functions. In addition, some software routines were designed for algorithms debugging. Finally, images visualization is done by means of custom software designed to work with the other two parts, and serves as an interface to manage the files of the captured images.



CONTENIDO

AGRADECIMIENTOS	IV
RESUMEN	V
ABSTRACT	VI
CONTENIDO	VII
INTRODUCCIÓN	1
CAPÍTULO 1: CAPTURA Y PREPROCESAMIENTO DE IMÁGENES	3
1.1 PERCEPCIÓN, CAPTURA Y DIGITALIZACIÓN DE IMÁGENES.....	3
1.2 PREPROCESAMIENTO DE IMÁGENES DIGITALES.....	6
1.3 FORMULACIÓN DEL PROBLEMA.....	6
1.4 OBJETIVOS DE LA TESIS.....	7
1.4.1 <i>Objetivo general</i>	7
1.4.2 <i>Objetivos específicos</i>	7
JUSTIFICACIÓN DE LA TESIS	8
CAPÍTULO 2: ESTADO DEL ARTE	9
2.1 SEÑALES DE VIDEO	9
2.1.1 <i>Exploración de imágenes (breve historia)</i>	9
2.1.2 <i>Tipos de señales de video</i>	11
2.1.3 <i>Señales digitales</i>	18
2.1.4 <i>La cámara de video como fuente de imágenes</i>	18
2.2 TIPOS DE IMÁGENES.....	20
2.3 MODELO MATEMÁTICO.....	21
2.4 FUNCIONES Y ALGORITMOS PARA EL PREPROCESAMIENTO DE IMÁGENES DIGITALES.....	24
2.4.1 <i>Funciones matemáticas básicas</i>	24
2.4.2 <i>Funciones lógicas</i>	24
2.4.3 <i>Umbralizado, Re-mapeo de niveles de gris, gradientes y tablas de referencia.</i>	24
2.4.4 <i>Funciones geométricas</i>	25
2.4.5 <i>Funciones estadísticas y de probabilidad</i>	25
2.4.6 <i>Convolución y correlación espacial</i>	26
CAPÍTULO 3: SISTEMAS ACTUALES Y SOLUCIÓN PROPUESTA	27

3.1 DIGITALIZACIÓN DE IMÁGENES.....	27
3.2 PREPROCESAMIENTO DE IMÁGENES	29
3.3 SISTEMAS CONJUNTOS DE CAPTURA Y PROCESAMIENTO	30
3.4 VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS ACTUALES.....	32
3.5 PROPUESTA PARA CAPTURAR Y PREPROCESAR IMÁGENES DIGITALES	32
3.6 VENTAJAS DEL SISTEMA PROPUESTO	33
3.7 FORMA EN LA QUE SE RESOLVERÁ EL PROBLEMA DE DIGITALIZAR, PREPROCESAR UNA IMAGEN Y MOSTRAR EL RESULTADO.....	33
CAPÍTULO 4: CAPTURA DE IMÁGENES (DIGITALIZACIÓN).....	35
4.1 PLANEACIÓN.....	37
4.2 LOS DISPOSITIVOS Y COMPONENTES	38
4.3 EL DISEÑO	39
4.3.1 Decodificación de video.....	39
4.3.2 Memoria de datos.....	40
4.3.3 Lógica de control.....	41
4.3.4 Circuito impreso.....	44
CAPÍTULO 5: PREPROCESAMIENTO DE IMÁGENES	45
5.1 IMPLEMENTACIÓN	46
5.2 ALGORITMOS	47
5.3 ADECUACIÓN DE RESULTADOS	51
5.4 SISTEMAS DE PREPROCESAMIENTO Y COMPONENTES	53
5.4.1 La computadora personal como sistema de preprocesamiento	54
5.4.2 Sistema de preprocesamiento basado en un procesador de señales digital	55
CAPÍTULO 6: VISUALIZACIÓN DE RESULTADOS.....	57
6.1 REQUERIMIENTOS DEL PROGRAMA DE VISUALIZACIÓN	57
6.2 EL PROGRAMA	58
6.2.1 Estructura de archivos del programa	60
6.2.2 Presentación y características del programa	61
CAPÍTULO 7: RESULTADOS Y CONCLUSIONES	65
7.1 OBTENCIÓN DE RESULTADOS	65
7.1.1 Captura de imágenes.....	65
7.1.2 Preprocesamiento de imágenes.....	66
7.1.3 Visualización	70
7.2 CONCLUSIONES, METAS Y LOGROS ALCANZADOS	70
7.2.1 Circuito de captura	71
7.2.2 Preprocesamiento de imágenes.....	71
7.2.3 Visualización	72
7.3 MEJORAS Y TRABAJOS A FUTURO.....	72
BIBLIOGRAFÍA.....	73
APÉNDICE.....	76

Lista de Figuras  y tablas 

1.1 Sistema de captura de imágenes humano.....	3
1.2 Sistema artificial de captura de imágenes.....	3
1.3 Calidad de la imagen a diferente número de muestras.....	4
1.4 Calidad de la imagen a diferentes tamaños de bits por píxel.....	5
1.5 Diagrama básico de digitalización de imágenes para señales de video.....	5
1.6 Preproceso que se realizan buscando ciertas características.....	6
2.1 Forma de la señal de video compuesto NTSC.....	9
2.2 Señal de video RS-170.....	10
2.3 Señal de video RS-170A NTSC.....	10
2.4 Componentes del video RGB.....	11
2.5 Composición del video RGB analógico.....	12
2.6 Composición del video RGB digital.....	12
2.7 Señal de sincronía en el color verde.....	13
2.8 Componentes diferenciales de video.....	13
2.9 Señal de video S-VHS o S-video.....	14
2.10 Ventana o mascara de 3x3 píxeles.....	20
2.1, Valores numéricos y cantidad de bits para las diferentes tonalidades de grises.....	17
2.2, Valores numéricos y cantidad de bits para representar los diferentes colores tipo RGB.....	17
2.3, Valores numéricos y cantidad de bits para representar los diferentes colores tipo YCbCr.....	18
3.1 Frame Grabber Model 311 de Sensorat.....	23
3.2 IMQ PCI-140/PXI-1407 de National Instruments.....	23
3.3 FlashBus MV Lite de Integral Technologies.....	24
3.4 A436TM Parallel Video DSP Chip de OMD.....	24
3.5 Video procesador mondream de ARVOO.....	25
3.6 SMT309 Advanced Imaging Module de SUNDANCE.....	25
3.7 SMT314 Video Frame and Display C40 TIM de SUNDANCE.....	25
3.8 Estación de administración de imágenes digitales de I ³	26
3.9 Esquema básico del sistema de captura y preprocesamiento de imágenes propuesto.....	28
4.1 Diagrama de flujo para la digitalización de imágenes provenientes de una señal de video compuesto.....	30
4.2 Diagrama de flujo para la digitalización de imágenes de video utilizando un Decodificador de Video.....	30
4.3 Relación de píxeles y líneas para diferentes dispositivos y formatos estándares de proyección....	31
4.4 Decodificación de la señal de video.....	33
4.5 Almacenamiento de la imagen en memoria.....	35
4.6 Diagrama de la lógica de control.....	35
4.1 Espacio de las Imágenes en Kilo Bytes.....	34
5.1 Diagrama de flujo del sistema de preprocesamiento de imágenes digitales.....	45
5.2 Conjunto ordenado que conforma los datos en la imagen.....	46
5.3 Pasos para la aplicación del algoritmo de preprocesamiento a las imágenes.....	46
5.4 Perspectiva plana de una imagen en espacio esférico.....	48
5.5 Esquema para determinar las distancias ArcX y ArcY.....	49
5.6 Forma de la imagen final al trasladarse de una superficie esférica a una plana.....	49
5.7 Manejo de la posición del píxel cuando sus valores no son enteros absolutos.....	51
5.8 Combinación de valores de píxel en la misma posición.....	52

5.9 Acotamiento de los valores de un píxel.....	52
5.10 Ajuste al tipo de dato manejado en los píxeles.....	52
5.11 Componentes básicas de un sistema de preprocesamiento.....	53
5.12 Operación del modulo de preprocesamiento en la PC.....	54
5.13 Procesador de señales digital.....	55
5.14 Operación del módulo de preprocesamiento con el sistema basado en el procesador de señales digital o DSP.....	56
6.1 Diagrama de flujo del programa de visualización de resultados.....	57
6.2 Modelo de objetos del programa de visualización.....	58
6.3 Esquema de operación del programa de visualización.....	59
6.4 Fólder de inicialización del programa de visualización.....	61
6.5 Fólder de salida, encargada del formato de salida de la imagen.....	62
6.6 Fólder de configuración, modifica diversos parámetros de la decodificación.....	62
6.7 Control del programa, modifica el modo de operación del decodificador.....	63
6.8 Fólder de captura de imágenes.....	63
6.9 Fólder de algoritmos de preprocesamiento.....	64
6.10 Ventanas de administración, visualización y de parámetros de preprocesamiento.....	64
7.1 y 7.2 Imágenes provenientes de una mini cámara de vigilancia que muestra al propio prototipo realizando la captura y a un par de objetos sobre un patrón geométrico.....	65
7.3 Imagen capturada de televisión publica.....	66
7.4 Imagen capturada de televisión restringida.....	66
7.5 y 7.6 Imágenes de películas capturadas desde un reproductor de DVD.....	66
7.7 Imagen capturada original 1.....	67
7.8 Imagen binarizada con un umbral de niveles de gris igual de 137.....	67
7.9 Función NOT sobre la figura 7.7.....	67
7.10 Función Máximo entre las figuras 7.7 y 7.8.....	67
7.11 Función Mínimo entre las figuras 7.7 y 7.8.....	67
7.12 Función Resta entre las figuras 7.7 y 7.8.....	67
7.13 Función Suma Absoluta entre las figuras 7.7 y 7.8.....	68
7.14 Función Suma Promedio entre las figuras 7.7 y 7.8.....	68
7.15 Función AND entre las figuras 7.7 y 7.8.....	68
7.16 Función OR entre las figuras 7.7 y 7.8.....	68
7.17 Función XOR entre las figuras 7.7 y 7.8.....	68
7.18 Imagen capturada original 2.....	69
7.19 Transformación de una superficie esférica a una plana.....	69
7.20 Imagen capturada original 3.....	69
7.21 Convolución espacial entre la figura 7.20 y la ventana 7.1.....	69
7.22 Convolución espacial entre la figura 7.20 y la ventana 7.2.....	70
A.1 Decodificador de video.....	77
A.2 Memoria de almacenamiento de video.....	78
A.3 Circuito de control y comunicación de datos.....	79
A.4 Circuito lógico de la escritura a los módulos de memoria.....	80
A.5 Circuito lógico de control e interfaz con el puerto de la PC.....	81
A.6 Circuito lógico del modulo I ² C.....	82
A.7 Circuito impreso 1.....	83
A.8 Circuito impreso 2.....	83

INTRODUCCIÓN



Este documento tiene como finalidad servir de guía para diseñar un sistema que permite la captura de imágenes provenientes de cualquier fuente estándar de video, darle un formato digital, aplicar un algoritmo de preprocesamiento que garantice un nivel mínimo de calidad y mostrar el resultado. El sistema tendrá la finalidad de proporcionar imágenes que puedan ser útiles en tareas de tratamiento de imágenes, visión por computadora, reconocimiento de patrones, etc. Con esto en mente, se ha organizado la presente tesis en un conjunto de capítulos, los cuales se describen a continuación:

En el primer capítulo, se da una breve introducción a la captura y preprocesamiento de imágenes. Dentro del capítulo se definen los objetivos a lograr para proponer una solución que permita implementar algoritmos que mejoren o modifiquen imágenes con el fin de obtener resultados útiles en diferentes materias. También es aquí donde se da la justificación que motiva la realización de la presente tesis y de llevarla a cabo.

Dentro del capítulo dos, se aborda el estado del arte, los antecedentes y un ámbito más amplio de la teoría y el conocimiento de las señales, sistemas y estándares que son necesarias para poder desarrollar la tesis.

El capítulo tres da un panorama del ámbito que rodea al problema a resolver presentando ejemplos de sistemas y circuito que desempeñan funciones equivalentes o similares al sistema a realizar y establece una nueva propuesta de solución para resolver el problema, donde se incluyen características y ventajas diferentes a las ya existentes. También se describen los recursos de cómputo a utilizar y el diseño de pruebas de funcionamiento a aplicar.

Los capítulos cuatro, cinco y seis describen las partes en que se desarrolla la solución del problema. Se presenta una explicación del diseño, su funcionamiento y los elementos de hardware y programación desarrollados, así como su interacción con las partes complementarias. Las partes en las que se compone la solución son: Captura de imágenes (capítulo 4), Preprocesamiento de imágenes (capítulo 5) y Visualización de resultados (capítulo 6) que constituyen la parte práctica de la tesis.

El último capítulo contiene los resultados que explican el desempeño del diseño propuesto al aplicar pruebas de funcionamiento para los casos que el sistema puede resolver. Se describen los resultados generados y se mide la eficacia del diseño. Además, de estos resultados se realizan las conclusiones que se comparan con los objetivos planteados al inicio de la tesis tanto en lo general como en los casos particulares y se dan las observaciones y mejoramiento a futuro del diseño construido.

Finalmente, se presenta la bibliografía consultada para la elaboración de la tesis y un anexo digital en disco compacto conteniendo los documentos, archivos, un glosario de términos usados, esquemas, imágenes y demás material utilizado para la realización de la tesis y construcción del prototipo.

CAPÍTULO 1: CAPTURA Y PREPROCESAMIENTO DE IMÁGENES

1

1.1 Percepción, captura y digitalización de imágenes

Desde la perspectiva humana, a la representación de la realidad que es percibida en forma de luz, capturada, traducida y descompuesta por el sentido de la vista y luego decodificada e interpretada por el cerebro, se le denomina imagen.

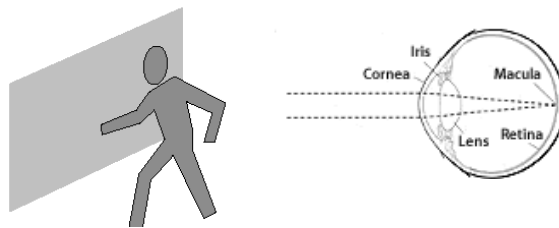


Fig. 1.1 Sistema de captura de imágenes humano.

A diferencia del cerebro que puede descomponer y procesar los elementos que conforman una imagen (como es la intensidad de luz y el color) al mismo tiempo, una máquina tiene que manejar cada elemento que la compone de forma separada y trasladarla a una dimensión que pueda manejarse (por ejemplo un plano); por lo tanto, para capturar una imagen se tiene que realizar una exploración (para tomar todas las muestras posibles dentro de una área determinada dándoles un orden adecuado) que se denominará ‘escaneo’ y recoger toda la información posible que la conforma.

De manera similar se han desarrollado dispositivos que capturan esta misma realidad en forma de información que luego es almacenada en papel, celuloide, medios magnéticos, mecánicos, eléctricos, etc. Para el interés de la tesis se requiere que la información a manejar sea de forma digital, por lo que los dispositivos que nos permiten esto son el ‘Escáner’, la cámara fotográfica digital y la cámara de video.

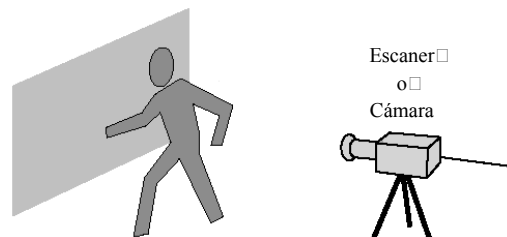


Fig. 1.2 Sistema artificial de captura de imágenes.

En contraposición con la cámara fotográfica o el escáner que toman toda la imagen y el resultado lo depositan en un medio físico (como la memoria no volátil o discos magnéticos), la cámara de video genera como salida convencional una señal eléctrica denominada ‘señal de video compuesto’ que contiene la información de la imagen de forma secuencial, esto es, en el mismo orden que se realiza la exploración o escaneo de la imagen. Debido al modo y lo económico que es generar la señal de video, una cámara de video nos permite con cierta facilidad realizar la captura y digitalización en la misma forma secuencial en que es transmitida.

Para digitalizar una imagen se requiere de dos parámetros: la cantidad de muestras y el valor de profundidad, que desde este momento lo llamaremos por el término conocido como ‘**píxel**’, ya sea en niveles de gris o en colores. Dependiendo de la cantidad de los píxeles que se tomen se obtendrá una calidad determinada; esta decisión se toma en el diseño ya que para obtener una imagen aceptable se debe seleccionar una cantidad determinada de píxeles así como el tamaño de cada uno de ellos, esto indicará la cantidad de recursos necesarios para poder llevar a cabo la construcción del sistema. Además se debe considerar el uso o aplicación donde se utilizará la imagen, así como la propia fisiología del ojo humano.

El sentido de la vista se caracteriza por retener las imágenes percibidas en un breve instante, suficiente para no ver cambios entre un cuadro y otro de una transmisión de cine o televisión, además el ojo tiene un grado finito mínimo de distinción de nitidez, contraste, brillo y color en una imagen que se mira. Valiéndose de esto se ha demostrado que en términos prácticos una imagen de 128 x 128 píxeles (tomando escalas en potencias de 2 por compatibilidad digital) es suficiente para que el ojo humano no pueda percibir cambios entre un píxel y el siguiente en una imagen con profundidad o resolución constante.

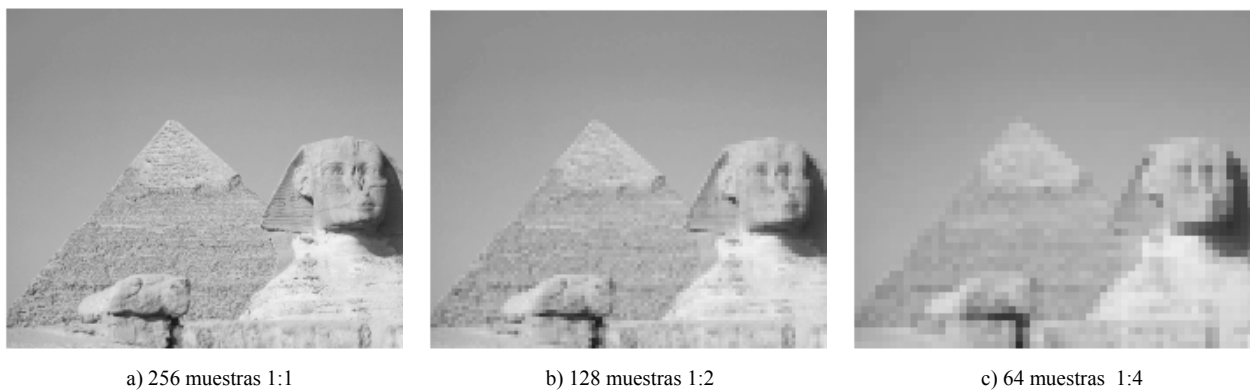


Fig. 1.3 Calidad de la imagen a diferente número de muestras.

Por otro lado el valor de profundidad mínimo de cada píxel es menos específico ya que depende de la imagen que se trate y de la aplicación en que se utilice, cuando se trata de procesos que requieren reconocer la forma más que el detalle se necesita un grado mínimo de profundidad; sin embargo, para aplicaciones donde se necesite resaltar los detalles, los valores deben ser preferentemente los más cercanos al valor analógico real. También es en este punto donde se debe de considerar el manejo en color o en blanco y negro, determinado por la aplicación en que se utilizará.

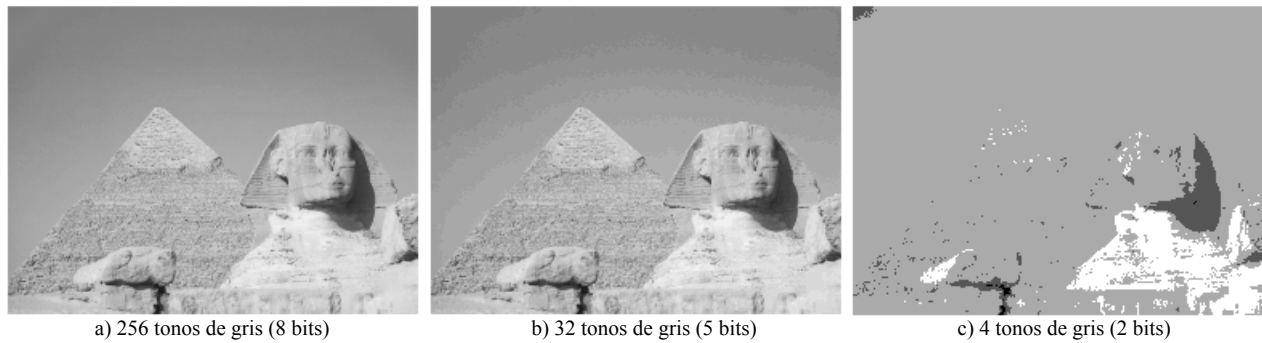


Fig. 1.4 Calidad de la imagen a diferentes tamaños de bits por píxel.

Una vez que se tienen contemplada la cantidad de píxeles y el tamaño de éstos, se procede a realizar la digitalización que para el caso de esta tesis se obtendrá de una señal de video (de manera similar a un receptor de televisión); para esto se describe a continuación un esquema básico que nos muestra cómo se convierte dicha señal en datos digitales que posteriormente se utilizarán para su preprocesamiento.

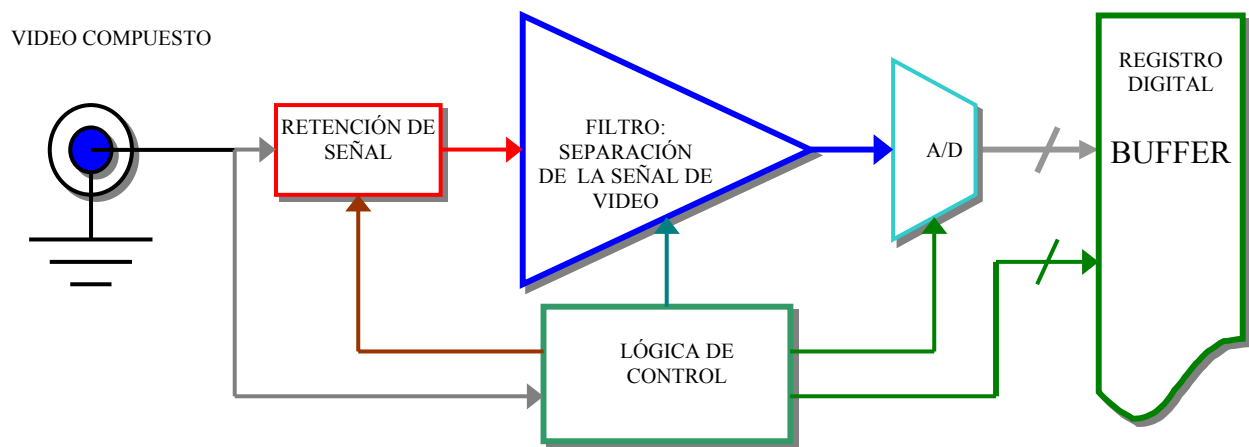


Fig. 1.5 Diagrama básico de digitalización de imágenes para señales de video.

De cualquier fuente de video se filtra la señal con el fin de dejar la información que conforma la imagen, si se requiere se obtiene la información que corresponde al color separándola de la señal principal. La mayor parte de la información corresponde a la luminosidad que se utiliza para generar imágenes en blanco y negro. Al realizar la conversión analógico–digital se obtiene la información en forma de datos digitales (píxeles) y se almacenan en un medio adecuado. Es importante destacar que dependiendo de la velocidad con que se realice la conversión y el tamaño (número de bits) de los píxeles, determinarán la resolución y el tamaño de la imagen. Finalmente los píxeles almacenados tienen un orden y una posición que permite tener una representación de la imagen capturada; a este conjunto se le denomina como **‘imagen digital’**.

1.2 Preprocesamiento de imágenes digitales

Diferentes aplicaciones requieren de imágenes, en una gran variedad de casos se necesita que tenga un determinado grado de procesamiento, éste puede ser muy sencillo o de alta especialización como: el filtrado (que permite eliminar ruido o realzar bordes), el umbralizado (donde se puede modificar los niveles de intensidad), la binarización, etc. En ocasiones se aplica uno o varios procesos en combinación para un resultado en particular. Aunque existen diversas opiniones y puntos de vista sobre el punto en que una imagen es ‘procesada’ o ‘preprocesada’; Se ha determinado identificar a todo aquel proceso que se realice sobre una imagen y que éste no sea utilizado por una aplicación en particular se le denominara desde este momento como **‘Preprocesamiento de la imagen’**.



Fig. 1.6 Preproceso que se realizan buscando ciertas características.

Entre las cosas que se pueden realizar en el preprocesamiento están los cambios de contraste, brillo, color, ampliar, reducir, resaltar bordes, eliminación de ruido etc. como se ve en la figura 1.6. La naturaleza bidimensional de las imágenes determina que muchas de estas funciones sean algoritmos que se aplican sobre matrices y sus operaciones. Por otra parte hay funciones lineales que se aplican sobre el valor de cada píxel, como por ejemplo la asignación del número de colores o la cuantificación de ellos; para poder realizar esta tarea se tienen la necesidad de tener un conjunto de funciones que las realicen. Las funciones que se requieran pueden ser implementadas por herramientas de software sofisticadas o dispositivos electrónicos que contengan uno o varios algoritmos. Otra forma es realizar estas funciones a la medida; de esta forma pueden implementarse algoritmos dentro de dispositivos electrónicos que sirvan para construir circuitos y sistemas que puedan ser integrados para aplicaciones dedicadas pero estos se caracterizan por ser muy caros y de alta especialización. Ejemplos de estos se ven en el capítulo tres, referente a los sistemas y métodos actuales.

1.3 Formulación del problema

Cada vez más existen diferentes tareas en las que se requiere de imágenes; en gran medida es necesario que estas tengan un preprocesamiento previo, por lo tanto, se planea diseñar y construir un sistema que permita implementar uno o varios algoritmos básicos de preprocesamiento. También se pretende que este diseño sirva como una herramienta que permita probar diferentes funciones y comparar los resultados con respecto a la imagen original. Con esto se pretende dar paso a la generación de nuevas ideas y proyectos específicos en distintas áreas. Por lo tanto se plantea: **¿Cómo construir un sistema que capture una imagen en forma digital, implemente un preprocesamiento sobre ésta y entregue como resultado una nueva imagen y que se pueda utilizar como herramienta básica para diferentes aplicaciones?**

1.4 Objetivos de la tesis

Basándose en lo anterior, los objetivos de la tesis son:

1.4.1 Objetivo general

Diseñar un sistema que capture, implemente un algoritmo de preprocesamiento y entregue el resultado en forma de una nueva imagen o datos que puedan ser útiles para un determinado propósito o interés en distintas áreas del conocimiento y la ingeniería.

1.4.2 Objetivos específicos

a) Construir un sistema que capture una imagen proveniente de cualquier fuente de video compuesto que cumpla con los estándares NTSC o PAL (cámara de video, VCR, etc. los dos tipos de señales más utilizadas en el mundo), convierta esta imagen en formato binario y ofrezca esta información a su salida.

b) Diseñar e implementar en un sistema conveniente de procesamiento digital, algoritmos de preprocesamiento de imágenes preseleccionados. Este sistema utilizará la información proveniente del circuito de captura, aplicará el algoritmo y entregará un resultado en forma de una nueva imagen o datos digitales.

c) Realizar un programa de interfaz que permita visualizar el resultado entregado por los sistemas de captura y preprocesamiento. Este programa podrá mostrar la imagen original y su resultado para comparar y evaluar el desempeño del algoritmo utilizado.

Justificación de la tesis

Los sistemas que procesan imágenes son de gran especialización y de un alto costo de implementación, pero por otro lado su uso es cada vez más frecuente. Se tiene la necesidad de construir sistemas especializados, de acuerdo a aplicaciones específicas o particulares. Para esto el construir un sistema para implementar un algoritmo, o procedimiento ideado para una aplicación en particular se facilita si el sistema de procesamiento digital es capaz de ser reprogramado con el fin de realizar ajustes en el mismo, mejorar el resultado, o bien para observar el efecto que produce al realizar cambios en los parámetros y valores que alteran la imagen original. De lo anterior se puede decir que lo que se pretende es construir una herramienta básica que ayuda a capturar una imagen, aplicarle un algoritmo de preprocesamiento, y mostrar el resultado para compararlo con la imagen original y que este resultado sea útil en las áreas del procesamiento de imágenes.

Si bien el tema no es totalmente innovador a nivel mundial, nacional o regional, tiene su interés, por tratarse de tecnologías relativamente nuevas en nuestro ámbito. Por otro lado, existe un especial interés en que ésta tesis pueda ser de utilidad en futuros proyectos y trabajos de índole similar.

CAPÍTULO 2: ESTADO DEL ARTE

(ÁMBITO Y ANTECEDENTES)

2

2.1 Señales de video

2.1.1 Exploración de imágenes (breve historia)

La forma en que se explora o escanea una imagen en video es la misma de la televisión. Los fundamentos principales de la televisión fueron descubiertos en el siglo XIX. En 1884 Paul Nipkow obtuvo una patente por un método opto-mecánico de transmisión de imágenes: el disco Nipkow, una rueda con 24 agujeros pequeños a lo largo de una línea espiral. Éste permitía extraer la imagen con 24 muestras o puntos en una sola línea conformando un ‘cuadro’ y transmitirla línea por línea.

A partir de los descubrimientos e invención de los tubos electrónicos y por ende del tubo de rayos catódicos, en 1911 se propuso por parte de A. Campbell Swinton un proyecto de televisión electrónica que utilizara el tubo de rayos catódicos para captar y reproducir las imágenes, el rayo desviado magnéticamente podría realizar una exploración más veloz que la de los sistemas mecánicos. El primer tubo electrónico de cámara de Televisión fue inventado por Vladimir Zworykin en 1923, el “iconoscope”, éste contenía en su parte más ancha una ‘placa-mosaico’ (lámina de mica cubiertas de células fotoeléctricas) y en su parte contraria un cañón de electrones con bobinas deflectoras para barrer en ‘**zigzag**’ toda la placa. El rayo variaba su intensidad según la carga de la célula, lo que determinaba una variación de la corriente que emitía la placa-mosaico, a la que se le llamó señal de video. El número de líneas barridas fue variando con el tiempo, la primera transmisión práctica de la televisión fueron realizadas por John Logie Baird en 1926 que usaba 30 líneas por cuadro y 5 cuadros por segundo, después fueron 120 líneas, para 1935 se alcanzaron 343 líneas y luego 403. La BBC comenzó las primeras transmisiones de la televisión regular con un barrido de 405 línea por cuadro con 25 cuadros por segundo en 1936, usado hasta 1945. La segunda guerra mundial detuvo el desarrollo de la televisión, en cambio ayudo a la invención del radar. Después de la guerra muchos otros países comenzaron transmisiones de la televisión. Francia usó un sistema de 819 líneas por muchos años, pero la mayoría de los países europeos escogieron un sistema de 625 líneas. En EE.UU. fue un sistema de 525 líneas denominado RS-170, que todavía está en uso.

EE.UU. fue el primero en usar color, en 1953. Japón fue el segundo, en 1960 usando el sistema americano de transmisión de televisión NTSC (National Television Systems Committee). En 1965 Europa no adoptó este sistema: Francia utilizó el SECAM (SEquentiel A Memoire ("memory

sequential")), Europa oriental tomo una variación de SECAM, mientras la mayor parte del resto de Europa decidió usar el PAL (Phase Alternation Line) que finalmente paso a ser su estándar.

En la actualidad, las imágenes son exploradas o escaneadas conformando un área rectangular. El barrido o escaneo se realiza de forma lineal y que puede ir de izquierda a derecha, de arriba abajo o en alguna otra dirección con el fin de recorrer toda el área. Una vez convertidas en una señal de video estas mismas imágenes se encuentran contenidas en tramas de señal que se transmiten 30 veces por segundo. Cada trama en si, esta conformada de un número finito de porciones iguales que contienen la información de una "línea o renglón" de la imagen. Este número finito de líneas es una limitación de la señal de video (525 líneas para el caso de NTSC), por lo que la información contenida en cada línea debe de ser coincidente para tener una imagen bien proporcionada.

Además de la información de la imagen que puede ser en blanco y negro o a colores denominadas como "luminancia" y "crominancia" respectivamente, la trama de la señal de video contiene un conjunto de pulso que se denominan como "señal de sincronía" y que son las instrucciones para reconstruir la imagen en un receptor (la televisión); estos pulsos sirven en este caso para conocer el comienzo y final de una imagen, una línea y referencias para la identificación del color. Como resultado de la segmentación de la información, una imagen se define por conceptos como "cuadro (frame)" que es la instancia de la imagen completa, "campos (fields)" par e impar ya que en la mayoría de los casos las líneas que contienen la información están intercaladas en dos grupos a diferentes tiempos, etc. Como referencia práctica, en el video compuesto NTSC RS-170A (Fig. 2.1) la señal de video es la suma de 3 señales: sincronía (SYNC), luminancia (LUMA) y crominancia (CHROMA) referidas al reloj de la subportadora maestra de 3.579545 MHz para su sincronía; cada línea de la imagen consta de: un pulso de sincronía horizontal que va de 0 a -0.4 V que indica cuando termina y comienza una línea, un estallido de color o 'Color Burst' que se compone de 9 a 11 ciclos de subportadora con amplitud de -0.2 a 0.2 V que es usado como referencia de la crominancia para extraer el color de acuerdo al desplazamiento de la fase, finalmente se tiene un nivel de luminancia con un umbral que va de 75mV para el negro a un máximo de 1V para el blanco; si es el caso, la señal de croma se modula por fase para la transmisión del color. A la suma de las señales de luminancia, crominancia t sincronía se le denomina como señal de "video compuesto".

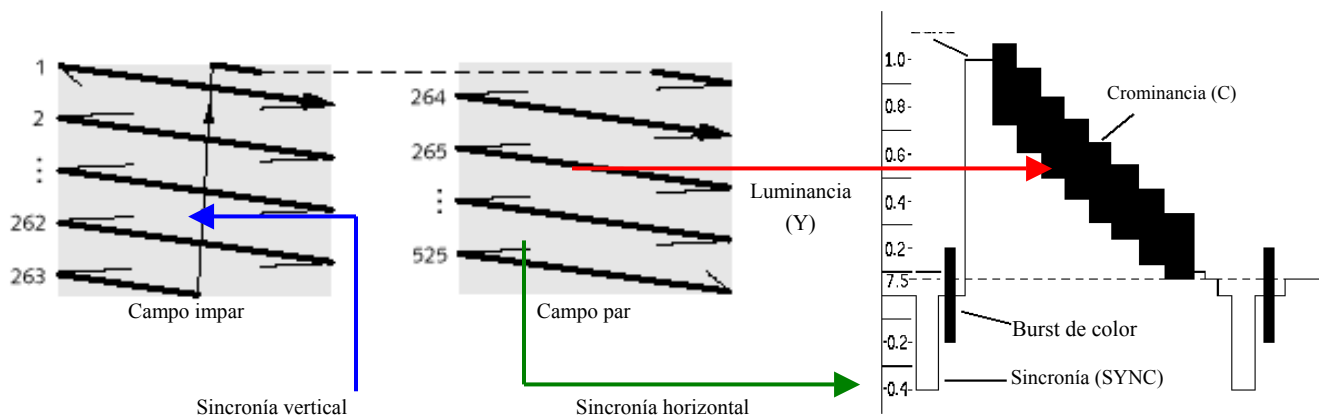


Fig. 2.1 Forma de la señal de video compuesto NTSC.

En el mundo existen varios modelos de señal de video, a continuación se describen algunos de los más usados. Si se desea conocer más sobre la conformación de la señal de video, así como de la codificación y decodificación puede consultar las referencias en CD y enlaces de Internet en [Anexo\1) Captura de imágenes, Anexo\1) Captura de imágenes\Páginas web y Anexo\1) Captura de imágenes\Enlaces web]

2.1.2 Tipos de señales de video

Video monocromático RS-170 (RS-170 Monochrome Video Signal).

El estándar RS-170 es propuesto por la asociación de industrias electrónicas de los EE.UU. (Electronic Industries Association EIA) para la transmisión de video en blanco y negro (B&N). Éste fue el primer estándar desarrollado en los primeros años la televisión. La señal RS-170 consiste de dos componentes: luminancia (luminance) Y sincronía (sync). Esta señal es fácil de procesar y transmitir. La luminancia porta la información de la imagen en blanco y negro y su detalle. La brillantez de la imagen varia con respecto al nivel de voltaje. La SINCRONÍA porta las señales de sincronización vertical y horizontal y es adicionada a la de luminancia, figura. 2.2.

Video compuesto RS-170A NTSC (RS-170A (NTSC) Composite Video).

El estándar RS-170A es propuesto por la EIA para la transmisión de video a color. Este estándar fue adoptado como una modificación del RS-170 con leves cambios en las frecuencias de sincronía (sync) y adicionando una señal subportadora de 3.58MHz para codificar y portar la información de color o crominancia (chrominance). La señal RS-170A se constituye de tres señales: luminancia (luminance o luma), sincronía (sync) y una tercera denominada crominancia (chrominance o chroma). Todas las partes de la señal son referenciadas al reloj maestro de la subportadora con una frecuencia exacta de 3.579545 MHz. Una parte de la señal de crominancia denominada ‘estallido de color o color burst’ se usa para referenciar la crominancia y obtener la información del corrimiento en la de fase y con esto, extraer la información de color en un sistema de televisión. Las tres señales (sync, luma, chroma y color burst) están sincronizadas con respecto al reloj maestro de la subportadora, figura 2.3.

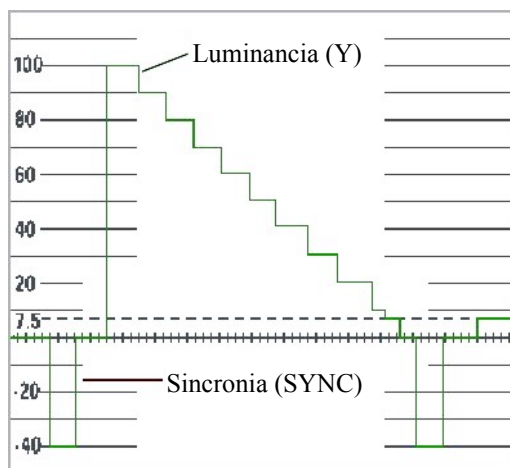


Fig. 2.2 Señal de video RS-170.

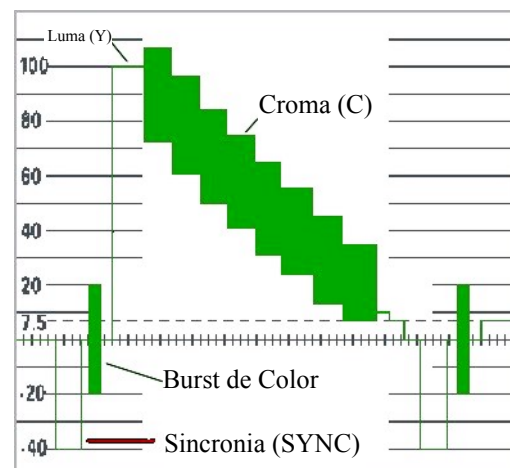


Fig. 2.3 Señal de video RS-170A NTSC.

Video compuesto PAL (PAL Composite Video)

El sistema de televisión a color denominado PAL (Phase Alternating Line) que describe el método de intercambio en 180° de la fase del 'color burst' en la alternancia del escaneo de las líneas, tiene la ventaja de cancelar el error resultante en el color proveniente de la distorsión entre la amplitud y la fase durante la codificación del color. EL sistema PAL también tiene mejor resolución que el NTSC; con 625 líneas y mejoras en el ancho de banda de los canales ROJO y AZUL debido a una mayor frecuencia en la subportadora de 4.43 MHz.

La mayor frecuencia en la subportadora produce menos interferencia y arrastre de puntos (dot crawl) en la imagen que el NTSC ya que la subportadora es justo el ancho de banda de la señal de LUMINANCIA en los equipos que usan PAL.

La codificación PAL también integra la necesidad de un control de tonalidad o matiz en los equipos dando una reproducción de color más verdadero, que no posee el NTSC.

La principal desventaja del PAL es un refresco vertical de 50Hz para cada campo y de 25Hz para cada cuadro; esta frecuencia es algo baja, que hace que el ojo humano puede ver un parpadeo (flicker) en la imagen. Una variación de este sistema es el PAL-M que tiene las características de 525 líneas, como el NTSC pero usa la misma frecuencia de subportadora para la codificación (4.43MHz) y por lo tanto es diferente al NTSC4.43.

Video compuesto SECAM (SECAM Composite Video)

EL sistema SECAM que viene del acrónimo: Sequential Coulrur Avec Memorie (sequential color with memory) o color secuencial con memoria es una señal compleja, diseñada para sobrellevar o superar todos los problemas de ruido e interferencia que tiene el NTSC. Fue diseñado como adición a la señal de blanco y negro de televisión de Europa para mantener compatibilidad con los equipos viejos existentes.

El SECAM usa un sistema único de doble subportadora que llevan la información de los colores ROJO y AZUL en líneas secuenciales. Para obtener el dato completo de color estas dos líneas tienen que ser comparadas. Esto requiere de salvar o retener la línea previamente transmitida en una memoria o en un retenedor o retraso analógico para así poder realizar la comparación y determinar el color. Para aumentar su inmunidad al ruido el SECAM usa una técnica de modulación en FM al contrario del AM del NTSC y PAL. Por su alto costo y complejidad no a sido muy utilizado.

Componentes del video RGB (RGB Component Video)

El RGB es la base principal de todo el video a color y de cualquier señal de video compuesto generada. Por ejemplo, la cámara de TV profesional usa 3 sensores de imágenes separados: uno para la luz ROJA, para la VERDE y para la AZUL. Cada sensor tiene su filtro apropiado enfrente para eliminar los otros colores.

Para su distribución (vía aérea, cable o por satélite) las tres señales son codificadas en una sola denominada video compuesto como el NTSC, el PAL o el SECAM. Esta señal compuesta es entonces enviada a su destino y ahí es decodificada en un monitor de TV que maneja los tres rayos de electrones del tubo de proyección de la TV (cinescopio).

El proceso de hacer el color a partir del RGB es la ADICIÓN, esto es, que las señales de cada color ROJO, VERDE y AZUL son sumadas para obtener la imagen completa.

Anatomía del RGB: en la figura 2.4 se muestra un esquema donde los canales ROJO, VERDE y AZUL componen un patrón de barras. Note los cambios en los bordes de cada color, figura 2.4.

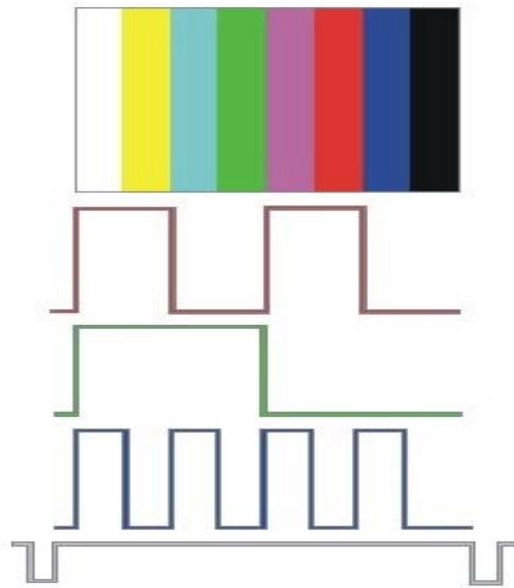


Fig. 2.4 Componentes del video RGB.

Video RGB analógico (Analog RGB Video)

La señal RGB analógica esta compuesta de cuatro señales: señal ROJA (RED), señal VERDE (GREEN), señal AZUL (BLUE) y la señal de sincronía compuesta (SYNC COMPOSITE).

Todas las señales son diseñadas para ser distribuidas vía cable coaxial de 75 Ohms con una carga de terminación de 75 Ohms. Cuando esta apropiadamente terminada, el video RGB debe de tener un valor máximo de 0,697 volts (0.7V) DC, mientras el signo de SYNC COMPOSITE es usualmente negativo, 0.4 volts en amplitud pero variaría dependiendo del equipo generador de las señales. A diferencia del RGB DIGITAL que está en 'ON' o en 'OFF', el RGB analógico puede tener un infinito número de niveles de voltaje y estados que proporcionan una reproducción verdadera del color, figura 2.5.

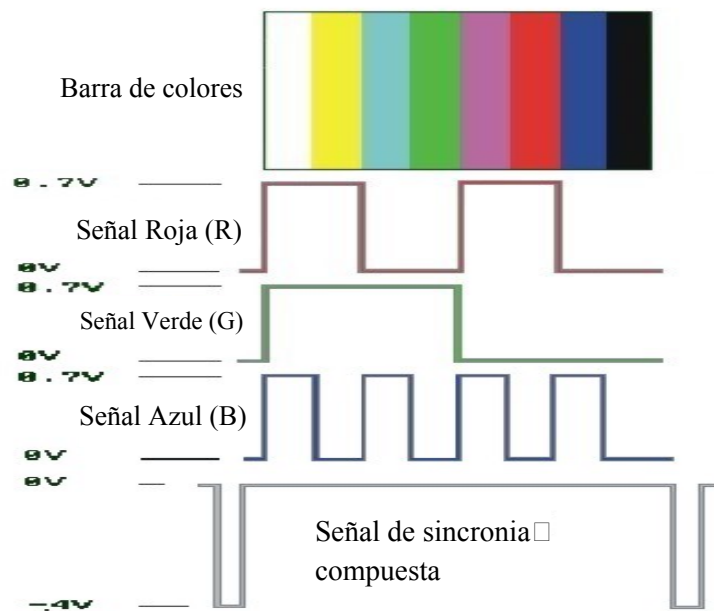


Fig. 2.5 Composición del video RGB analógico.

Video RGB digital (también llamado RGBI).

Es usado en monitores de computadoras de bajo costo tal como la IBM-PC con un monitor denominado CGA (Adaptador Gráficos de Color). El CGA usaba 4 bits de memoria para determinar los colores con un total de 16 posibles combinaciones. Los tres canales RGB expanden los 8 colores posibles debido al cuarto bit, que se llama canal de INTENSIDAD (I). Esto cambia el brillo global del monitor CGA de color de 50% a 100% de saturación. Viendo el ejemplo se nota cómo el canal de intensidad cambia el brillo. En diferencia con el RGB analógico, el RGB digital es puesto en ON o OFF, de tal manera que no hay otros estados intermedios para hacer otros colores. Todas las señales son estados lógicos TTL donde es cero volts para OFF y cinco volts para ON.

Las señales de sincronía horizontales y Verticales son proporcionadas por el monitor CGA y también son niveles TTL. En sistemas gráficos de computadora de 24 bits, cada canal tiene 8 bits, que hacen una posibilidad de 2 a la octava potencia de combinaciones de color (256) para cada señal R G y B de canal de color. La cantidad de combinaciones posibles con 24 bits es de: $256 \times 256 \times 256 = 16.8$ millones de colores, figura 2.6.

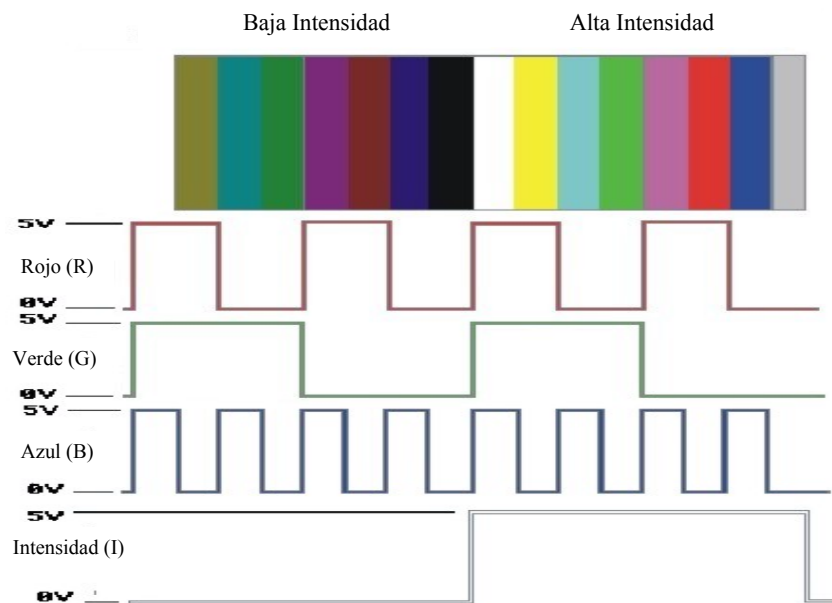


Fig. 2.6 Composición del video RGB digital.

Video con sincronía en el verde del RGB (Sync on Green RGB Video)

EL Video con sincronía en verde del RGB es idéntica al RGB analógico, excepto que las señales de sincronización están sobrepuestas en el canal VERDE. Esto reduce a tres el número de cables requeridos para llevar la señal. Todas estas señales son diseñadas para transmitirse vía cable coaxial de 75 Ohms y terminadores de la misma impedancia. Cuando es así, la señal de video tiene un valor de pico de 0.697 volts (aproximadamente 0.7v) CD, mientras que la señal de sincronización compuesta es usualmente negativa con una amplitud de 0.4 volts pero es muy dependiente del equipo. Este tipo de RGB no puede ser utilizado por todos los equipos y monitores, algunos equipos profesionales pueden ser configurados para aceptarlo, estos equipos extraen la componente de sincronización sobre puesta en el canal VERDE. Se puede notar que el canal verde en realidad llega a ser en efecto una señal de video compuesto RS-170 monocromática, y puede ser vista en cualquier monitor de blanco y negro o de color con solo conectarla o enchufarla a la entrada de video de cualquier equipo que interprete video compuesto. Profesionalmente esta señal se usa como 'señal clave (Key signal)' para efectos de video, figura 2.7.

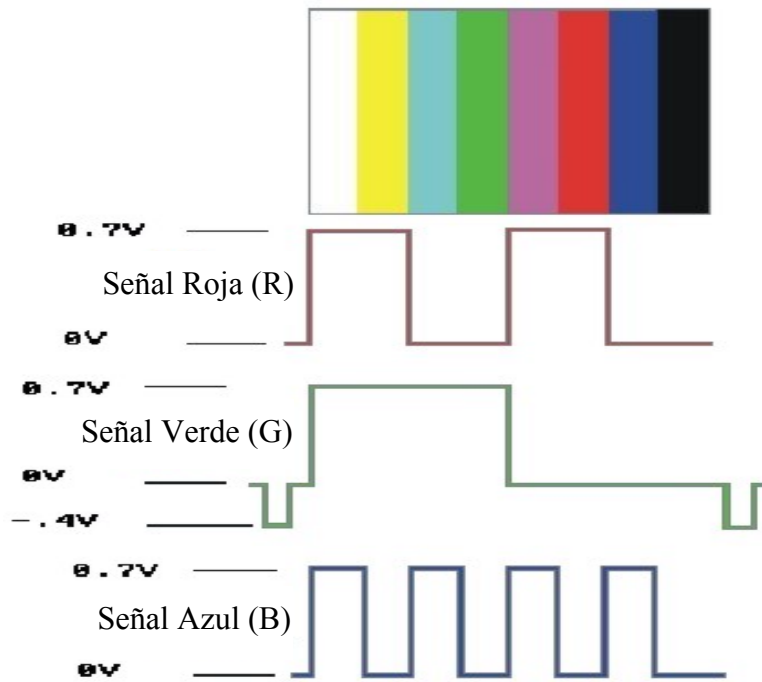


Fig. 2.7 Señal de sincronía en el color verde.

Componentes Y, R-Y y B-Y de video (R-Y/B-Y/Y Component Video Signal).

Este estándar de video se ha venido usando solo desde hace pocos años, aunque el concepto ya había sido considerado desde el comienzo de la TV a color. En efecto muchos sistemas de televisión utilizan este tipo de señales internamente para transferir el color entre los circuitos.

La corporación Sony obtuvo el crédito por impulsar estas señales como un mecanismo útil de transferencia de medios con el avance de las máquinas de grabación en cinta llamadas BETACAM (no debe confundirse con el sistema comercial betamax). Este sistema fue el primero que pudo grabar una señal de color en cinta separando las componentes de la señal de color. Sony seleccionó R-Y, B-Y y Y en lugar de RGB ya que estas nuevas señales tienen un reducido ancho de banda que les permite ser grabadas en cinta en formulaciones de tiempo. Esto también resulta fácil de convertir a y desde NTSC ya que hubo una gran disponibilidad de circuitos integrados en los equipos de TV, figura 2.8.

Ecuación de video R-Y/B-Y/Y

$$Y = 0.59G + 0.30R + 0.11B \quad (\text{eq. 2.1})$$

$$R-Y = 0.70R - 0.59G - 0.11B \quad (\text{eq. 2.2})$$

$$B-Y = 0.89B - 0.59G - 0.30R \quad (\text{eq. 2.3})$$

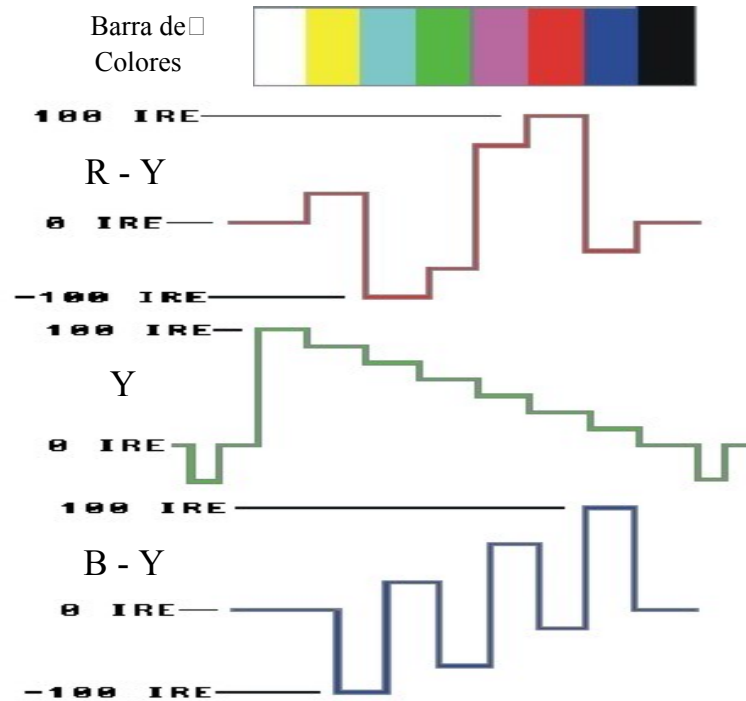


Fig. 2.8 Componentes diferenciales de video.

Como se puede ver de la ecuación estas señales dan al canal Y el ancho de banda más grande mientras reducen el ancho de banda del ROJO y el AZUL. El arreglo de ancho de banda de las tres señales es: 4MHz - 2MHz - 2MHz, esto es a menudo llamado video 4:2:2 (en referencia a los anchos de banda) y es excelente para cintas ya que los problemas de la codificación en NTSC no son introducidos en la señal. El 4:2:2 a llegado a ser rápidamente el estándar de la producción en video profesional. El diagrama en el dibujo muestra como las tres señales (R-Y, B-Y y Y) aparecerían en un sistema de video como la grabadora de cinta BETACAM de Sony.

Estas señales se colocan de tal forma que correspondan a la barra de colores estándar. Nótese que como son señales de diferente color, estas pueden tener niveles de voltaje negativos, mientras que la señal NTSC compuesta por la adición de estas es siempre positiva. Nótese también que la posición de la señal Y porta las señales de video compuesto sobrepuesta en ella, similar a la de video con sincronía en verde (sync green).

Componentes del video S-VHS (Y/C (S-VHS) Component Video).

El video S-VHS (también llamado Y-C o S-Video) tiene la ventaja de una alta resolución y mejor reproducción de color y que se mantienen separados por canales diferentes la señal de LUMINANCIA (Y) y la señal de CROMINANCIA (C) de la señal NTSC original.

Por tal razón, las interferencias que normalmente se presentan en NTSC son eliminadas. Otra ventaja de esta separación es la de poder fácilmente duplicar la señal vía cinta, ya que la codificación y decodificación en los equipos de S-Video no necesitan ordenar la CROMINANCIA con respecto de la LUMINANCIA y esto también deja de introducir los usuales errores y distorsiones que vienen con cada generación de una cinta de video.

Los niveles de las señales de S-VHS son los mismos que los del RS-170A (1 volt pico) y cada porción esta diseñada para ser transmitida vía coaxial con sus respectivos terminadores e impedancias de 75 Ohms. En la figura 2.9 se muestra una comparación entre las señales de S-VHS y la de NTSC con respecto al esquema de barra de colores.

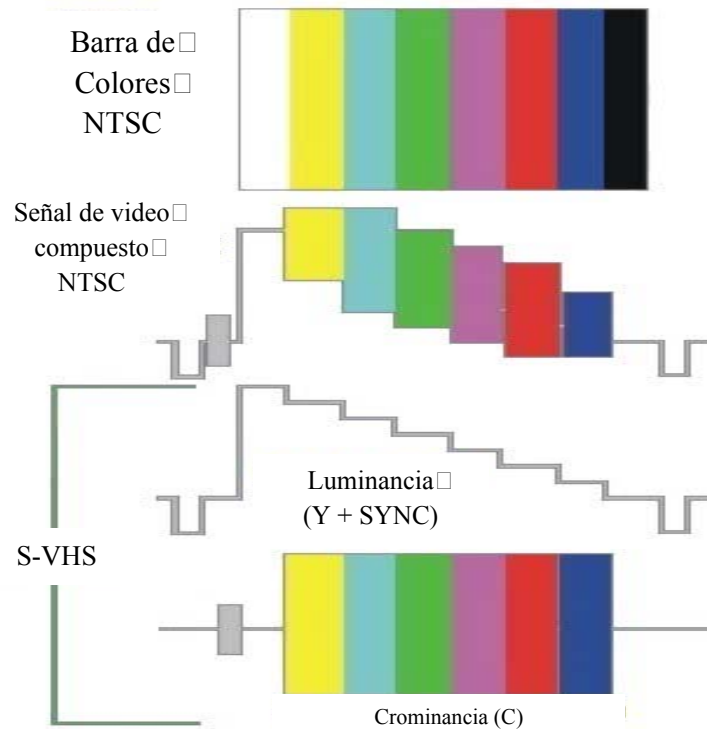


Fig. 2.9 Señal de video S-VHS o S-video.

El video se puede obtener en forma de señal eléctrica de cualquier fuente estándar, ya sea una cámara, reproductora o grabadora de video, etc. Dependiendo de la fuente y la aplicación que usará la imagen, se puede filtrar para limpiar de ruido o para eliminar componentes innecesarias. Los filtros más utilizados son regularmente analógicos y se utilizan para separar la información de luminancia y de crominancia [[Anexo\1\) Captura de imágenes\ NTSC Decoding Basics Part *.pdf](#)].

Tasas de cuadros por segundo y líneas por cuadro usado actualmente en sistemas de video y que son necesarios para el diseño y construcción del circuito de captura.

NTSC/ EE.UU., "sistema M":

525 líneas y 29,97 cuadros /seg.

B&N: frecuencia de cuadro = $f_v = 30$ Hz

Frecuencia de línea = $f_h = 525 * f_v = 15750$ Hz

Color: la frecuencia de cuadro y de línea tuvo que ser reducida aproximadamente 0,1% para hacer que la subportadora de sonido tenga una frecuencia de 4.5 MHz, un múltiplo par de la frecuencia de línea. Esto minimiza la interferencia de la señal de color hacia la de sonido. Esta modificación se realizó por que los televisores viejos no podían modificar la frecuencia de subportadora de sonido, sin embargo no tenían dificultad de sincronizar las frecuencias de cuadro y de línea modificadas (Nota: Este sistema es el utilizado en México).

Frecuencia de subportadora de sonido = $f_s = 4,5$ MHz

Frecuencia de línea = $f_h = f_s / 286 = 15734.2657343$ Hz

Frecuencia de cuadro = $f_v = f_h / 525 = 29.9700299701$ Hz

PAL CCIR/Europa, sistemas B y G

625 líneas y 25 cuadros/seg.

Frecuencia de cuadro = $f_v = 25$ Hz

Frecuencia de línea = $f_h = 625 * f_v = 15625$ Hz

Frecuencia de subportadora de sonido = $f_s = 352 * f_h = 5,5$ MHz

Desde la frecuencia de subportadora de sonido original ya era un múltiplo par, por lo que no se necesita un ajuste en las frecuencias de cuadro y línea al introducir el color.

Con respecto a otros dispositivos que también exploran el área de la imagen como es el escáner, tienen la desventaja de necesitar que ésta se encuentre capturada ya en otro medio físico como la fotografía.

La cámara de video tiene la ventaja de explorar y convertir en señal una imagen directamente del medio y de poder realizarlo de manera continua.

2.1.3 Señales digitales

Existen nuevas formas de manejar el video motivadas por el advenimiento de la televisión digital y de alta definición, el DVD y el VCD entre otros; en este caso se define un modelo de imágenes digitales combinado con un proceso de compresión, todo esto derivado de los grandes volúmenes de información que estos manejan. En este caso las imágenes ya están digitalizadas y tiene el formato JPEG, para la transmisión se comprimen los datos, esta compresión es basada en la transformación discreta cosenoidal y es la utilizada en el formato MPEG que además incluye los datos del sonido. Aun que es de gran interés manejar este tipo de señales, la disposición de dispositivos es todavía limitado. Por otra parte las señales analógicas manejadas actualmente serán vigentes por algún buen tiempo. Para saber más de este tipo de señales puede consultarse las referencias de Internet [URL9], o en el CD [Anexo\1) Captura de imágenes\Páginas web].

2.1.4 La cámara de video como fuente de imágenes

En la actualidad existen muchos tipos de cámaras, estos tienen circuitos y tecnología diversos para capturar la imagen; para diferenciarlas se deben de considerar características como: el tipo de escaneo, de interfaz, los colores que puede generar y las características del sensor, ya que de todos estos factores depende la calidad de la imagen que se genera; estas características se ven reflejadas en la señal eléctrica de video que será digitalizada. Para cada característica se puede tener las siguientes opciones:

Tipo de escaneo

- Escaneo de área entrelazada (Interlaced Area Scan) – Una cámara de escaneo de área entrelazada es aquella donde un cuadro de video está compuesto por dos campos entrelazados. El número de líneas en un campo es la mitad del número de líneas en un cuadro entrelazado.
- Escaneo lineal (Line Scan) – Una cámara que toma imágenes en una dimensión. Imágenes de dos dimensiones o 2D pueden ser compuestas por múltiples líneas.
- Escaneo de área progresivo (Progressive Area Scan) – Una cámara de escaneo progresivo es aquella donde el cuadro completo de video es expuesto al mismo tiempo.

Tipo de interfaz

- Analógico - una cámara que transforma la información de luz en píxeles y donde cada nivel de luz de cada píxel es convertido en una señal video analógico.
- LVDS - tecnología de señal diferenciada de bajo voltaje (Low Voltage Differential Signaling LVDS) (también conocida como RS-644) ofrece una señal diferenciada de baja tensión y rápidos tiempos de transición. Las interfaces de LVDS proporcionan un alto desempeño, bajo voltaje y poco ruido en la transmisión de datos.
- Channel Link – El canal de enlace (channel link) se basa en la tecnología de señal diferenciada de bajo voltaje (LVDS), el Channel Link proporciona interconexiones de alta velocidad y bajo ruido. El protocolo proporciona bajo ruido, alta velocidad de transferencia de datos en cables cortos y largos.
- Digital - una cámara que transforma la información de luz directamente a valores finitos denominados píxeles y donde cada uno de estos se maneja dentro y hacia fuera de la cámara.
- IEEE 1394, también conocido como 'FireWire', es un bus serial de alto rendimiento originalmente desarrollado por 'Apple computer' en los años 90. La especificación de la línea maneja índices de rendimiento de procesamiento de 100 Mbits/s, de 200 Mbits/s, y de 400 Mbits/s. El IEEE 1394 también ofrece la tecnología 'hot pluggable' que permite a los dispositivos ser conectada y desconectados mientras el sistema esta prendido.

Color

- Color CCD-3 (3-CCD Color) – Imagen de color producida por 3 arreglos independientes de sensores tipo CCD, uno para cada color primario rojo, verde y azul. Una imagen es pasada a través de un prisma que separa sus componentes roja, verde y azul. Cámaras 3-CCD producen imágenes claras y de una alta calidad de color, pero son muy caras.
- Monocromáticas (Monochrome) – Una cámara que produce imágenes en blanco y negro. Otras pueden ser que solo toman una componente de la luz, como verde, ultravioleta o infrarrojo.
- Color de Mosaico (Mosaic Color) – La imagen de color es producida por una cámara que usa un CCD y un esquema de codificación (como el algoritmo de Bayer) para separar el color. El resultado es usualmente una señal compuesta. Los Filtros de mosaico (Mosaic filters) hacen posible que las cámaras sean más pequeñas y baratas, pero la técnica sacrifica resolución (agudeza en la imagen) y sensibilidad a la luz.

Sensor

- Dispositivo de acoplamiento de carga (Charge-Coupled Device CCD). Dispositivo para imágenes de estado sólido que almacena una representación óptica de la imagen en forma de carga eléctrica producida por el efecto de la fotoconductividad. Un mecanismo de lectura convierte las cargas de imagen en una señal de video. Con sus diferentes formas y modificaciones, éste es el dispositivo masivamente utilizado hoy en día para la captura óptica de las imágenes.

2.2 Tipos de imágenes

Formatos de imágenes y píxeles.

Para poder manejar la imagen de forma digital, hay que convertirla en un formato conocido. Existen diferentes formatos en los que se puede manejar una imagen como es en Blanco y Negro, Escala de Grises, Color (en este último existen formatos como R-G-B, H-S-I, Y-Cb-Cr, etc).

Es importante saber el tipo de dato binario que se utilizara para representar cada píxel ya que de esto dependen factores como: La cantidad de memoria de almacenamiento, el tamaño de los registros y tipo de dispositivo que procesará los datos, ya sea una PC, un DSP, un Procesador, Microcontrolador, FPGA o arreglo de ellos. Entre los tamaños de los datos manejados tenemos:

Blanco y Negro o Escala de Grises: Este tipo de dato es muy simple y fácil de manipular, se tiene una sola componente que representa la intensidad, por lo que la cantidad de bits determina el número de los posibles valores que se pueden representar y a los que se le denominan ‘Niveles de gris’. La relación entre un nivel de gris y el valor binario se establece convencionalmente del 0 para negro al valor máximo para el blanco.

bits	Negro	Niveles de gris	Blanco
1	0	-	1
8	[0-15]	(1 - 254)	(240-255]

Tabla 2.1, Valores numéricos y cantidad de bits para las diferentes tonalidades de grises.

RGB de 8, 12, 16, 24 y 32 bits que están distribuidos en tres partes que representan la contribución de los colores Rojo (R), Verde (G) y Azul (B) en las siguientes proporciones:

RGB(bits)	ROJO(bits)	VERDE(bits)	AZUL(bits)
8	2	4	2
12	4	4	4
16	5	6	5
24	8	8	8
32	10	12	10

Tabla 2.2, Valores numéricos y cantidad de bits para representar los diferentes colores tipo RGB.

YCbCr que se compone de tres paquetes que representan la intensidad o información en blanco y negro (Y), la diferencia con respecto al color azul (Cb) y la diferencia con respecto al rojo (Cr), también se caracteriza por su forma de distribución, ya que mientras en RGB se entrega toda la información en un solo paquete, en YCbCr la componente de intensidad es invariable, pero la parte de croma o color esta dispuesta en paquetes de diferentes tamaños y en diferentes tiempos. La distribución de algunos tipos de paquetes YCbCr se presenta a continuación:

YCbCr (bits)	Y(bits)	Cb(bits)				Cr(bits)			
4:1:1 12	$8_{(t)}$	$2_{(t+3)}$	$2_{(t+2)}$	$2_{(t+1)}$	$2_{(t)}$	$2_{(t+3)}$	$2_{(t+2)}$	$2_{(t+1)}$	$2_{(t)}$
4:2:2 16	$8_{(t)}$	$8_{(t)}$				$8_{(t+1)}$			
4:4:4 24	$8_{(t)}$	$8_{(t)}$				$8_{(t)}$			

Tabla 2.3, Valores numéricos y cantidad de bits para representar los diferentes colores tipo YCbCr.

Existen muchos formatos más que se pueden utilizar, y cada uno de estos representa a un tipo de dato de imagen computacional, como sería el RGB para las imágenes en mapas de bits o archivos BMP, o el YCbCr que manejan los archivos JPG. Otros tipos son los TIFF, GIF, PCX/DCX, XIF etc.

Para el propósito de la tesis sólo se tomará en cuenta el tipo de archivo de mapas de bits (BMP) por su compatibilidad con la señal RGB y por su compatibilidad con las PC. El tipo de archivo JPG es otra alternativa, esta directamente relacionado con la señal de video de canales separados (YCbCr) usada para generar el video compuesto NTSC, PAL etc.; pero por practicidad solo se maneja el primero. Por otra parte, una vez teniendo un tipo de archivo, éste puede transformarse a otro utilizando diversos programas de software que se encuentran disponibles. Referente al formato del archivo, éste se describe más ampliamente en el anexo [Anexo3) Programa de visualización\Páginas web].

2.3 Modelo matemático

El formato obtenido por la captura y digitalización, produce un arreglo de valores numéricos finito ordenado en posiciones x_i, y_i denominados píxeles y que conforman la imagen. Un modelo sencillo se puede definir como una función bidimensional indicada por $f(x, y)$, donde el valor o amplitud de f en la posición (x, y) da las características o propiedades de la imagen en este punto. En el caso de una imagen monocromática este valor es la intensidad o nivel de iluminación de la imagen [2].

$$0 < f(x, y) < L \quad (eq.2.4)$$

donde el 0 es el valor más bajo que se puede representar y L el máximo valor, dando una escala de valores en un intervalo $[0, L]$.

Una vez obtenida la información esta se maneja como una matriz, donde cada elemento de está, representa una muestra o píxel que se captura como ha sido mencionado. El píxel o elemento de imagen es una componente que se constituye de un valor y una posición.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (eq.2.5)$$

A esta matriz de $N \times M$ elementos se le denomina como la **imagen digital** en donde se le pueden aplicar algoritmos matemáticos para la realización de un proceso. El algoritmo a aplicar puede ser: Un método espacial que usa una ventana o matriz denominada como una subimagen con dimensiones $W \times W$ donde W es impar preferentemente y la cual recorre toda la imagen manipulando directamente cada píxel; o una función que se compone de una determinada ecuación o fórmula que se aplica sobre $f(x,y)$ realizando transformaciones en diferentes dominios. Frecuentemente las funciones se aplican en el dominio de la frecuencia basadas en la transformada de Fourier. Finalmente el algoritmo entrega un determinado resultado que conforma una imagen nueva con características diferentes a la original.

En el caso de un método espacial la ventana se compone de diferentes pesos o valores distribuidos convenientemente dentro de la matriz y se seleccionan de acuerdo a la naturaleza del proceso a aplicar. Estos valores pueden determinar un filtro, un nivel de umbralizado o un gradiente, etc. Mientras que en el caso de una función esta puede ser: una suma, una resta, una cuantificación, un histograma, transformación etc.

La etapa de preprocesamiento nos permite dejar una imagen lista para utilizarse por otro dispositivo o sistema, esta puede depender o no del sistema que utilizará el resultado pero se busca en general: mejorar la calidad, resaltar detalles, o transformar en algo nuevo las imágenes.

Los métodos del dominio espacial tienen como base usar los valores de los píxeles (nivel de gris o colores) para conseguir mejorar la imagen. Desde un punto de vista matemático diremos que lo que queremos es ver que tipo de transformación se puede aplicar a una imagen para que el resultado sea otra imagen pero con características visuales diferentes; es decir queremos realizar transformaciones T que aplicadas a la imagen $f(x,y)$ nos dé otra imagen $\mathcal{G}(x,y)$ con nuevas propiedades visuales.

$$\mathcal{G}(x,y) = T(f(x,y)) \quad (eq.2.6)$$

Cuando se expresa que $\mathcal{G}(x,y)$ depende de $f(x,y)$ a través de T , se quiere establecer que el cálculo del valor de \mathcal{G} en el punto (x,y) depende en principio de todos los valores de $f(x,y)$.

En la realidad y usando la hipótesis de que el valor en un punto de la imagen tan solo esta relacionado con unos pocos píxeles que le rodean, entonces es que solo se consideran transformaciones T que están definidas sobre el píxel en (x,y) , los píxeles en el entorno (ver figura 2.10) que puede abarcar un área de 3×3 o mayor. Por entorno 3×3 de un píxel se consideran los píxeles más inmediatamente cerca de dicho píxel, es decir los píxeles marcados con una '*’.

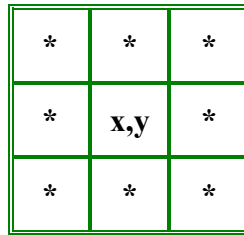


Fig. 2.10 Ventana o mascara de 3x3 píxeles.

Al conjunto de transformaciones que hacen uso de entornos para su definición las denominaremos Filtrado Espacial.

El fundamento de los métodos del dominio de frecuencias es el uso del teorema de convolución. La hipótesis de partida es que la imagen que queremos procesar, ha sido alterada por el mecanismo de captación de la imagen y sufrido una convolución con una determinada función que esta definida por las lentes del aparato de captura, el medio en que se toma la imagen o cualquier otro efecto, y que además esta contaminada por ruido aleatorio. La ecuación que define la imagen que resulta del anterior proceso es la siguiente,

$$\mathcal{G}(x,y) = h(x,y) * f(x,y) + \eta(x,y) \quad (eq.2.7)$$

Siendo $f(x,y)$ la imagen real, $h(x,y)$ la función de convolución que supondremos que es lineal e invariante respecto de la posición y $\eta(x,y)$ la función de ruido. Y el objetivo es calcular $\mathcal{G}(x,y)$ a partir de la anterior ecuación. De las dos posibilidades se ha optado por un método espacial, ya que se adapta convenientemente a la forma en que se maneja la captura de la imagen y por que podemos construir en el caso del filtrado, mascarar para el método espacial con especificaciones en el dominio de la frecuencia [2].

2.4 Funciones y algoritmos para el preprocesamiento de imágenes digitales

Por la naturaleza bidimensional de las imágenes, muchos de los algoritmos que se aplican son funciones matemáticas que se conforman de la siguiente manera:

2.4.1 Funciones matemáticas básicas

Entre éstas se tienen: operaciones aritméticas, que suman (+), restan (-), dividen (/) y multiplican (*) matrices. Suma, consistente en la adición del valor de cada píxel de la imagen, y cuyo valor se utiliza como factor escalar en histogramas y convolución basada en filtrado. Máximo y mínimo de píxel que determina el rango de valores usados en la imagen, otro máximo y mínimo es sobre la imagen en general, que entre un conjunto de imágenes, se toma ya sea la máxima o mínima como resultante, en el caso de trabajar con datos signados, se puede tener el absoluto. Negativo, consiste en que el valor de cada píxel es cambiado por su corresponde valor inverso, en el caso de trabajar con valores enteros sin signo y tamaño de 8 bits se tiene por ejemplo que, si el píxel tiene un valor de 255 su inverso es 0 definiendo una regla: píxel nuevo es igual al valor absoluto del píxel original menos el máximo valor que puede ser representado. Techo (*ceiling*) o piso (*floor*) de un píxel, que consiste en enviar al valor entero siguiente o inferior según el caso; ejemplo: si un píxel tiene valor 38.7 el techo es 39 y su piso 38. Módulo de dos valores, que consiste en el remanente después de sustraer el segundo valor del primero un número entero de veces; ejemplo: de 6.13 y 2.5, modulo es 1.13. Potencia y raíz de una imagen, consistente en la aplicación del valor potencia o raíz a cada píxel; funciones trigonométricas, exponenciales y logarítmicas, consistentes en aplicar la función correspondiente sobre cada píxel. Funciones con datos complejos, como la separación de su parte real e imaginaria o la de la magnitud y el conjugado. Las funciones matriciales, como la multiplicación de matrices, la inversión.

2.4.2 Funciones lógicas

Este tipo de funciones da resultados de cierto o falso y se puede representar como un valor (*0 o 1*) de estas funciones tenemos: Comparación, compuestas por, igual a ($=$), mayor que ($>$), menor que ($<$), mayor o igual (\geq), menor o igual (\leq), diferente (\neq), dando como resultado una imagen en donde cada píxel representa la verdad o falsedad de cada expresión, alguna (*any*) o siempre (*ever*) son funciones que se aplican sobre todo el conjunto de la imagen dando como resultado cierto o falso (un ejemplo es el de preguntar si algún píxel tiene valor a 45, si hay por lo menos uno, la respuesta es verdadero); operaciones lógicas entre bits, estas se llevan a cabo entre píxel y píxel conformándose por las operaciones *and*, *or*, *xor* y *not* el píxel resultante es el de la aplicación de la operación sobre cada bit del valor de éste; las mascarar, que consisten de una imagen del mismo tamaño que contiene información lógica o binaria para ser aplicada a otra imagen.

2.4.3 Umbralizado, Re-mapeo de niveles de gris, gradientes y tablas de referencia.

El umbralizado (Thresholding) es una operación donde el valor resultante de cada píxel depende de su correspondiente entrada, ya sea de un o más valores que se conocen con el nombre de umbral.

Éste se utiliza como un método simple de binarización y también se puede aplicar en segmentación ($R_i = \{H_i, I_i \geq T_i \text{ o } L_i, I_i < T_i\}$). Se puede hacer unbralización con diferentes criterios y valores de umbrales; el re-mapeo de niveles de gris es la forma de alterar o cambiar el brillo al sumar una constante (*bias*) y afectar el contraste multiplicando otra constante (*gain*) $R_i = \text{gain}I_i + \text{bias}$ o se puede cambiar el rango $I_{max} - I_{min}$ de los niveles de gris a otro rango $H - L$, $R_i = (I_i - I_{min})[H - L / I_{max} - I_{min}] + L$. Otra función que podemos aplicar es el gradiente o primera derivada, esta nos muestra los valores o partes de la imagen que cambian rápidamente, está se compone de una magnitud $m = ((\delta I / \delta x)^2 + (\delta I / \delta y)^2)^{1/2}$ y una dirección $\theta = \tan^{-1}(\delta I / \delta y / \delta I / \delta x)$, y se calcula mediante una aproximación $\delta I / \delta x = \lim_{\delta x \rightarrow 0} (I_{i+1} - I_i) / \delta x$, estas funciones pueden aplicarse más fácilmente utilizando la convolución espacial. Las tablas de referencia (look-up tables), son gráficas que indican de forma esquemática la manera en que cambian los valores de los píxeles, de esta manera podemos definir el umbralizado, el negativo, contraste de una imagen.

2.4.4 Funciones geométricas

Las funciones geométricas realizan cambios en el orden en que están colocados los píxeles; estas funciones son de: Espejo, que cambia el orden de las columnas invirtiéndolas. Espejo vertical, realiza la misma inversión, pero solo en los renglones. Transpuesta, que es la que cambia renglones por columnas. Rotaciones, que giran la imagen a la derecha o izquierda 90° , 180° o 270° y es similar a la función transpuesta. Interpolación, sirve para cambiar de tamaño una imagen y se basa en la eliminación de píxeles para reducir o en la de calcular píxeles intermedios para aumentar. Funciones de corte y pegado, donde se definen áreas que son copiadas o eliminadas para manejar solo una porción. Concatenación, donde se crea una imagen de la unión de dos o más. Rotación libre, esta definida por el cambio de la posición de los píxeles de acuerdo a la ecuación $Dx = dx \cos \theta + dy \sin \theta$ y $Dy = dy \cos \theta - dx \sin \theta$. Transformación de perspectiva, esta se compone del calculo de las coordenadas en 3D, Transformar estas en el espacio 2D de acuerdo a la proyección de la forma 3D en el plano estas juntas realizan la transformación. Finalmente una interpolación es necesaria para rellenar píxeles faltantes; desplazamiento (shifting) es otra función grafica que puede aplicarse a la imagen, consiste en trasladar una cierta distancia la posición de los píxeles tanto en x y/o en y .

2.4.5 Funciones estadísticas y de probabilidad

Estas funciones se utilizan para la reducción de datos. De las diferentes funciones, la media y la desviación estándar se pueden utilizar para discernir entre un conjunto de imágenes similares la de mayor brillantez o indicar que tan contrastante esta una. El cálculo esta dado por $\mu = 1/n \sum_{i=0}^{n-1} I_i$ para la media y $\sigma = [(1/n \sum_{i=0}^{n-1} I_i^2) - \mu^2]^{1/2}$ para la desviación estándar. Este tipo de operación se puede aplicar a una vecindad de píxeles donde hay un píxel central y sus vecinos próximos (8), de esta forma se pueden aplicar otras funciones como son el filtro mediano, filtro del vecino más próximo, el filtro sigma. Por parte de la probabilidad se tienen los histogramas que puede aplicarse en remapeo, equalización y como especificación.

2.4.6 Convolución y correlación espacial

Sin duda esta es la forma más sencilla de poder aplicar funciones matemáticas que de otra forma serían muy complejas y de un alto costo computacional.

Para esto se hace uso de una máscara que sirve como función de convolución h sobre una imagen I . Para realizar la convolución se calcula la suma de los productos entre los elementos de la máscara denominados pesos y los valores de los píxeles en el área que abarca la máscara, recorriendo sobre todos los elementos de la imagen. Los pesos contenidos en la máscara determinan el efecto de la convolución y por lo tanto define el filtro a aplicar. Este tipo de operación es de gran importancia práctica, ya que puede implementar una variedad diversa de filtros, por lo que es una de las principales herramientas para el preprocesamiento de imágenes.

Para una mayor referencia de funciones y algoritmos que pueden ser aplicados a una imagen referirse a [2] y [6] de la bibliografía.

CAPÍTULO 3: SISTEMAS ACTUALES Y SOLUCIÓN PROPUESTA

3

3.1 Digitalización de imágenes

Para esta tarea existen muchos sistemas ampliamente desarrollados. Cuentan con muchas características y opciones tales como la resolución de imagen, la cantidad de colores, su velocidad, etc. Pero también tienen particularidades que los limitan, ejemplo de esto es su compatibilidad con programas de software, su arquitectura eléctrica y por supuesto el costo [URL9]. De las diferentes opciones que se pueden encontrar mencionaremos las características más comunes en las que podemos referirnos para tener una muestra de los sistemas existentes:

Principalmente hay sistemas hechos en tarjetas con conexión de puerto PCI y últimamente USB, tienen una o más entradas de video de tipo BNC, S-Video y RCA para video-compuesto, s-video y RGB. En cuanto sus capacidades, las hay de baja y alta resolución que va de 320 x 240 a 1400x1050 o superior, estas pueden ser en blanco y negro o a color con una profundidad de píxel que va de los 8 a 32 bits. En su mayoría cuentan con un decodificador de video que reconoce diferentes sistemas de señal de video. El espacio en memoria regularmente es fijo, A continuación mostramos algunos ejemplos de estos sistemas.

‘Frame Grabber Model 311 de SENSORAY’ (Fig. 3.1).

Es un circuito integrado para la captura de señales que usa convertidores A/D tipo flash, con un manejo de señales PAL, SECAM, NTSC RS-170 CCIR y compatible, tiene puerto de control de cámaras, programación de contraste, brillo, Tonalidad y saturación; así como el uso de controladores (drivers) para ‘Windows’ propios para un puerto PC/104+ tipo bus. Con un costo de US\$ 250 es un convertidor adaptable a la señal de entrada y el procesamiento es dejado al software.

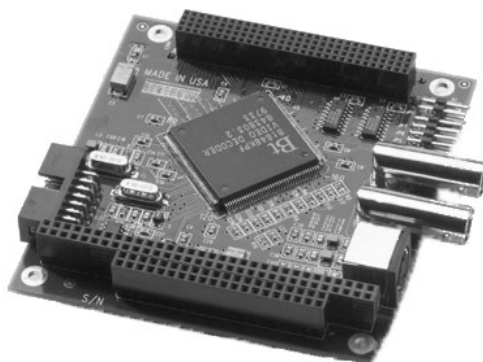


Fig. 3.1 Frame Grabber Model 311 de Sensorat.

‘IMAQ PCI-1407/PXI-1407 de National Instrument’ (Fig. 3.2).

Es una tarjeta de adquisición de imágenes mono canal en blanco y negro para señales NTSC y PAL, tiene una memoria FIFO para el almacenamiento, y cuenta preprocesamiento de escaneo parcial, programación de ganancia, desplazamiento (offset) y decimación. Con un costo de US\$ 765 a 985 es una tarjeta compatible con software propio de la marca y con una transferencia de datos en DMA para bus PCI.



Fig. 3.2 IMQ PCI-140/PXI-1407 de National Instruments.

‘FlashBus MV Lite de Integral Technologies’ (Fig. 3.3).

Es una tarjeta de adquisición dual para señales RS170, compuesta y Y/C; cuenta con un sistema de entradas y salidas para el control de cámaras o dispositivos y escaneo progresivo. Requiere de software propio del fabricante no proporcionado. El costo es de US\$ 336 más accesorios requeridos.



Fig. 3.3 FlashBus MV Lite de Integral Technologies.

3.2 Preprocesamiento de imágenes

Para la realización del preprocesamiento se tienen dos opciones, a través de programas de software o mediante el uso de un hardware dedicado; dado que en cualquier caso, usar un programa o utilizar un determinado dispositivo lo importante es el algoritmo, se considera a éste como la característica principal a considerar en los sistemas existentes. A diferencia de la gran cantidad de sistemas de captura existentes, los de preprocesamiento son muy pocos, de hecho, los sistemas que se encuentran en el mercado son realizados con tecnología de computación reconfigurable debido a la diversidad de algoritmos existentes para el preprocesamiento y de las diversas particularidades de las aplicaciones. La razón de esto es que es mejor reprogramar un solo dispositivo a construir uno para cada algoritmo.

Los sistemas disponibles se basan en un procesador de señales, un procesador dedicado o un construido a la medida, sus conexiones son únicas y en poca o gran medida se requiere de un sistema o programa de computo para utilizarlo. Los sistemas que se pueden encontrar tienen características muy avanzadas que garantizan la demanda de las aplicaciones en que se quieran utilizar, de esto entendemos que la mayoría son programables. El costo es un factor muy importante el cual determina su selección de acuerdo a las necesidades. De las opciones disponibles damos estos ejemplos:

‘A436™ Parallel Video DSP Chip de Oxford Micro Devices Inc’ (Fig. 3.4).

Es un procesador basado en tecnología RISC en paralelo y con 208 pines tiene entradas y salidas para diferentes sensores, puertos y señales de control, cuenta con una interfaz para programarse en lenguaje C de alto nivel y con una velocidad de 100MHz lo hace muy completo y versátil.



Fig. 3.4 A436™ Parallel Video DSP Chip de OMD.

Video Procesador ‘mondrean™ de ARVOO’ (Fig. 3.5).

Es un sistema paralelo basado en múltiples procesadores de señales, con una entrada de más de 24 bits de información y con salida analógica RGB, es un sistema potente para el manejo de video en tiempo real verdadero. Cuenta con 4 puertos externos de propósito general, JTAG programable y con poder de procesamiento de 5 Giga instrucciones por segundo lo hace capaz de realizar una gran variedad de tareas de alta complejidad. Con un costo de US\$ 2,948 es una opción muy cara.

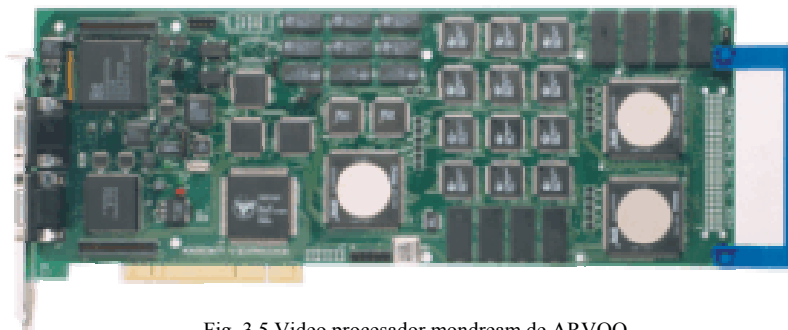


Fig. 3.5 Video procesador mondream de ARVOO.

3.3 Sistemas conjuntos de captura y procesamiento

Los sistemas que más se pueden encontrar, son aquellos que reúnen tanto la captura como el procesamiento; estos sistemas cuentan con características similares a las descritas antes para cada una. Lo que en verdad identifica a los sistemas que tienen la captura y el procesamiento juntos es el elevado costo que pueden alcanzar. A continuación veremos ejemplos de los que podemos encontrar:

‘SMT309 Advanced Imaging Module de SUNDANCE’ (Fig. 3.6).

Construido con base en el Video DSP paralelo de Oxford Micro Devices, 8Kbytes de memoria EEROM, 4Mbytes de trabajo, manejo de señal PAL NTSC SECAM y SVHS, bus de Interfaz global y salidas en CVBS o RGB. Se orienta a la compresión de video, es un sistema que puede ser reconfigurable, bajo requerimiento de procesamiento multicanal y efectiva reducción de tiempo de conversión. Con un costo de US\$ 2,875 es un sistema caro pero de amplias características.



Fig. 3.6 SMT309 Advanced Imaging Module de SUNDANCE.

‘SMT314 Video Frame Grabber and Display C40 TIM de SUNDANCE’ (Fig. 3.7).

Construido a base del procesador de señales TMS320C40 a 60MHz, 512 Kbytes de programa más 1MB adicional, 512Kbytes de EPROM, 4Mbytes de trabajo, manejo de señal PAL, resolución de 1024x512 no entrelazado y software 3L y Virtuoso de Daimond. Con un costo de US\$ 1,495 esta tarjeta diseñada para la captura y despliegue de imágenes PAL y optimizada para el procesamiento de señales esta orientada a sistemas de manejo de imágenes y gráficos, visión por computadora, reconocimiento de patrones, monitoreo de trafico y control de calidad.

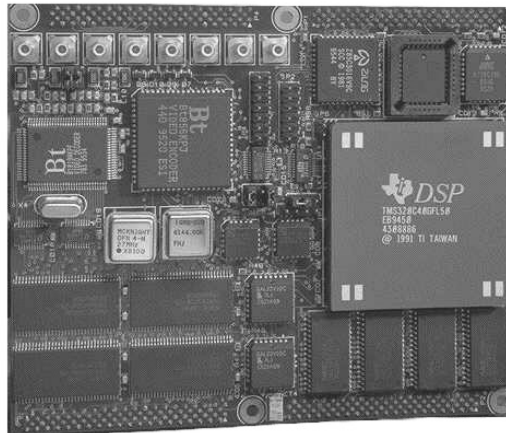


Fig. 3.7 SMT314 Video Frame and Display C40 TIM de SUNDANCE.

‘Estación de administración de imágenes digitales de I³’ (Fig. 3.8).

Es un sistema poderoso que adquiere, administra, almacena y actúa como servidor de imágenes digitales, utiliza programas de software para el procesamiento en una computadora PC, dispone de una tarjeta de captura de video de gran desempeño (alguna como las consultadas o mostradas aquí) y produce imágenes compatibles para su transmisión o distribución. Con un costo para la opción más sencilla de US\$ 995 resulta ser una solución muy limitada, para obtener resultados equiparables a la de los ejemplos anteriores el costo se eleva hasta en un 200%.

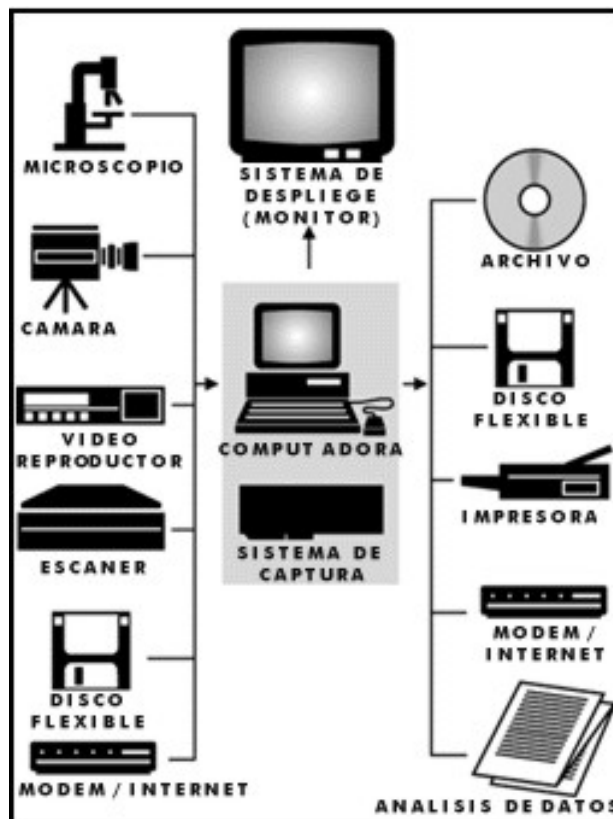


Fig. 3.8 Estación de administración de imágenes digitales de I³.

3.4 Ventajas y desventajas de los sistemas actuales

De las aplicaciones, sistemas y circuitos revisados y de los descritos en esta tesis, se obtienen una serie de ventajas y desventajas que destacaran las características que servirán para proponer el sistema a realizar.

Ventajas:

- + Sistemas con buena eficiencia y desempeño.
- + Los sistemas tienen flexibilidad al ser compatible con una gran cantidad de equipo.
- + Aplicaciones, procesos especiales y algoritmos depurados.
- + Uso de tecnología de vanguardia y un alto desempeño.

Desventajas:

- Sistemas exclusivos del fabricante y de alto costo.
- Tecnología de difícil disponibilidad en México.
- Algoritmos fijos que no se pueden cambiar o que es opcional a un mayor costo.
- Orientados solo a una o algunas partes del sistema y aplicaciones.
- El costo de los sistemas.

3.5 Propuesta para capturar y preprocesar imágenes digitales

Con base en los dispositivos existentes y el problema a solucionar, se propone diseñar un sistema digital y de software que se compone de 3 partes independientes y que puedan complementarse entre sí. Cada parte se describe a continuación:

Parte 1.- Circuito que captura imágenes de una fuente de 'video compuesto' del sistema M/NTSC (estándar de señal de video utilizado en México) que podrá ser proporcionado por cualquier fuente estándar como es una video cámara o video grabadora y un banco de memoria que almacene la imagen en forma digital.

Parte 2.- Un sistema de preprocesamiento independiente basado en algún procesador que permita implementar el algoritmo. Dada la variedad de opciones que se pueden utilizar y para respetar el criterio de independencia, se decide usar un sistema de desarrollo ya construido con el objetivo de dar mayor énfasis en la programación del algoritmo y el paso de los datos respecto al diseño del circuito eléctrico del sistema de preprocesamiento.

Parte 3.- Un programa de software que muestre las imágenes capturadas y los resultados para comparar y verificar la aplicación del algoritmo. El programa manejará las imágenes como archivos y proporcionará un archivo de imagen compatible a otros programas de software. El programa extraerá los datos del sistema de captura y generará un archivo de datos para que puedan ser enviados al sistema de preprocesamiento.

3.6 Ventajas del sistema propuesto

Con respecto a las ventajas del sistema propuesto para resolver el problema planteado se prevé, que el sistema a diseñar será modular, de tal forma que cada parte que lo componen pueda usarse de forma individual, entre dos de sus partes y con otras aplicaciones; También se pretende que sea escalable, para que puedan ser ampliadas sus capacidades como sería la inclusión de otros algoritmos de preprocesamiento, incluir la captura de video en tiempo real y el manejo de diferentes tipos de imagen digital; Además se pretende que el diseño pueda servir para generar soluciones a aplicaciones específicas que lleven a la construcción de nuevos circuitos y sobre todo a un costo menor.

La limitante en este diseño es la interfaz hacia la computadora personal y las capacidades de procesamiento, pero la intención de esta tesis es en principio permitir realizar pruebas y experimentar con diferentes imágenes y algoritmos. Por esta razón se deja como una de las mejoras a futuro la inclusión de estas características; esta y otras mejoras se describen en la sección 7.3 del capítulo de resultados y conclusiones.

3.7 Forma en la que se resolverá el problema de digitalizar, preprocesar una imagen y mostrar el resultado

La propuesta es: Capturar o extraer una imagen, proveniente de una señal de video compuesto (de acuerdo al estándar M/NTSC RS- 170^a establecido en México), entregar la imagen en formato binario para su preprocesamiento y mostrar el resultado de manera visual.

Se propone la construcción de un digitalizador, que capture un campo ($1/2$ cuadro de video) ya sea par o impar de tamaño fijo y almacenar el conjunto de píxeles que componen la imagen en un buffer o banco de memoria, para esto se utilizara un dispositivo conocido como 'Decodificador de Video' que integra las etapas de descomposición, filtrado y conversión digital de la señal de video compuesto, además proporciona el conjunto de señales necesarias para la manipulación de los píxeles. Con la ayuda de un dispositivo programable se implementa la parte digital de programación del decodificador y del control de la memoria. El tipo de memoria es FIFO, ya que es de gran utilidad para la transferencia de los datos.

Como tarea primordial se diseñará y programará el algoritmo de preprocesamiento en un sistema de desarrollo ya construido. Este sistema se programará mediante el uso de herramientas de software con algoritmos de preprocesamiento previamente seleccionados que contenga las características que se desean obtener de acuerdo a los objetivos de la tesis. De igual forma podrá modificarse para en lo subsiguiente agregar, rediseñar y optimizar más algoritmos que enriquezcan las capacidades de preprocesamiento.

Para la parte visual, se construye un programa de software creado en lenguaje C, visual y orientado a objetos para hacer la presentación de las imágenes originales y las que ya han sido transformadas en una computadora y que también pueda servir como herramienta para almacenarlas y también poder exportarlas al formato mapa de bits (BMP) que es compatible con otros programas de software.

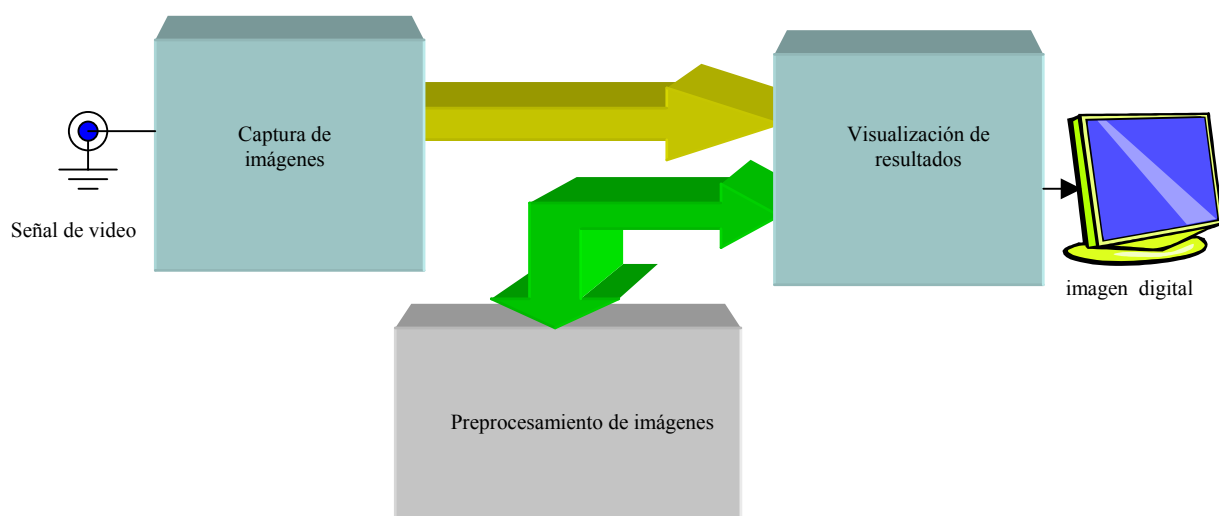


Fig. 3.9 Esquema básico del sistema de captura y preprocesamiento de imágenes propuesto.

El preprocesamiento de imágenes se conforma de un conjunto de algoritmos que realizan funciones primitivas específicas y reglas que gobiernen o manejen circunstancias excepcionales que se presenten, un ejemplo de esto es el caso en que el valor del píxel es cero y se le aplica la función logaritmo. Además se tiene que tomar en cuenta el tipo de datos y las restricciones de entrada. De lo anterior definimos una serie de reglas que sirven para diseñar los algoritmos que se implementarán y así conformar una biblioteca que será el sistema de preprocesamiento en esta tesis.

- Todas las imágenes serán del mismo tamaño: 244 líneas y 360 píxeles por línea.
- El dato que se maneja será de enteros de 8 bits sin signo (valores de 0 a 255) para niveles de grises, donde 0 corresponde a negro, y 255 a blanco.
- Los algoritmos a implementar serán basados en el método espacial, otro de los algoritmos a utilizar es el de tipo geométrico que es útil para corregir deformaciones por las lentes de la cámara. No obstante el sistema será capaz de utilizarse para programar una gran variedad de nuevas funciones que apliquen los algoritmos descritos en la sección 4 del capítulo 2 y muchos más que pueden ser adicionados posteriormente.
- La máscara será de tamaño fijo, 3x3 y 5x5. Se deja como opción posterior si se define una flexibilidad en el tamaño.
- Cuando toda parte de una máscara salga del área de la imagen los datos contenidos se tomarán como cero.

CAPÍTULO 4: CAPTURA DE IMÁGENES (DIGITALIZACIÓN)

4

En este capítulo se habla de las posibles formas de realizar la digitalización, los puntos que se toman en consideración en el diseño para convertir la señal a datos digitales, y una descripción del sistema de captura y digitalización de imágenes desarrollado para la tesis.

Para extraer la información que pasará a ser los píxeles que componen la imagen se tienen que identificar las diferentes componentes de la señal de sincronía como son la sincronía vertical para saber cuando comienza y termina un ‘campo (field)’, la sincronía horizontal que nos indica el inicio y final de una línea, los pulsos de blanqueo que son los espacios que no contienen información pero que sirven para indicar la posición de la señal en un televisor o monitor, señal par e impar que indica cual de los campos esta presente, el disparo o ‘burst’ de color y el nivel de luminosidad. Para ver en detalle todo lo referente a la composición de la señal de video consultar la bibliografía [3] y [5] o la documentación que se ofrece en el CD [Anexo\1) Captura de imágenes\Páginas web]. Para esto se puede construir un circuito analógico o utilizar alguno disponible en el mercado.

Para construir por cuenta propia todo el proceso de captura y digitalización se tiene que diseñar todos los filtros y detectores para identificar la información valida de las demás componentes. El proceso de captura se puede describir de la siguiente forma: Una etapa que acopla y retiene la señal de video, filtros que separan las señales de luminancia y crominancia, por medio del circuito de lógica de control se decodifica la señal de sincronía y se identifican los momentos correspondientes a la información de la imagen. Con estas señales se activa el muestreo en el convertidor analógico–digital que entrega valores digitales que corresponden a los niveles de voltaje que a su vez representan niveles de luminosidad de la imagen. Finalmente el control puede manejar el almacenamiento de los datos de salida. Para esto puede utilizarse el LM1881[URL3] que es un circuito que recibe la señal compuesta y entrega pulsos y señales que son útiles para el control y proceso de la digitalización. Como parte de la etapa de control es un circuito mixto que como salida entrega un pulso denominado ODD/EVEN para identificar el campo que se encuentra en ese momento; SYNC es otra señal que nos indica cuando tomar la información de luminancia y de crominancia. Para las otras operaciones se requiere de un filtro IRE que separe la componente de luminancia, por otra parte la crominancia es demodulada en fase. La digitalización se realiza por convertidores analógico–digitales A/D, uno en el caso de blanco y negro y hasta tres en el caso de color. Finalmente el resultado puede emplearse en línea o almacenado en forma de dato digital para su uso posterior. El almacenamiento puede ser temporal o permanente y debe ser lo bastante amplio para contener el total de muestras que compone la imagen a digitalizar (Fig. 4.1).

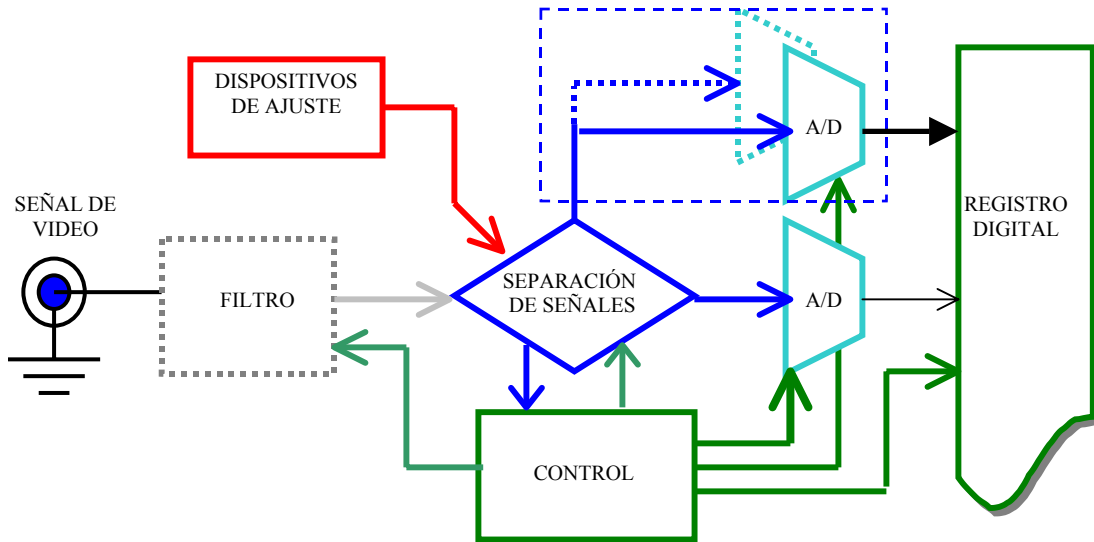


Fig. 4.1 Diagrama de flujo para la digitalización de imágenes provenientes de una señal de video compuesto.

Entre los dispositivos de alta integración que nos permiten implementar una digitalización más completa con un mínimo de componentes y con mayores características están el TVP5031 (TVP5022) [URL5] o el S5D0127X01 (KS0127 y KS0122) [URL4] que integran las etapas de captura y digitalización. Al utilizar estos dispositivos se facilita la tarea pero trae consigo nuevos enfoques de diseño que involucra un mayor uso de tecnología en la planeación, construcción e implementación del circuito y de los componentes, un ejemplo de esto es el control la programación que se tiene que realizar por medio del protocolo I²C [URL1] o en el CD [Anexo\1] **Captura de imágenes\Documentación de dispositivos**] usado convencionalmente para este tipo de dispositivos electrónicos. El funcionamiento es muy práctico, ya que al conectarse con todos los componentes requeridos solo se tiene que agregar la etapa para el control de escritura a memoria y de la programación interna del propio dispositivo, reduciendo notablemente el tamaño total del circuito y reduciendo el número de conexiones eléctricas.

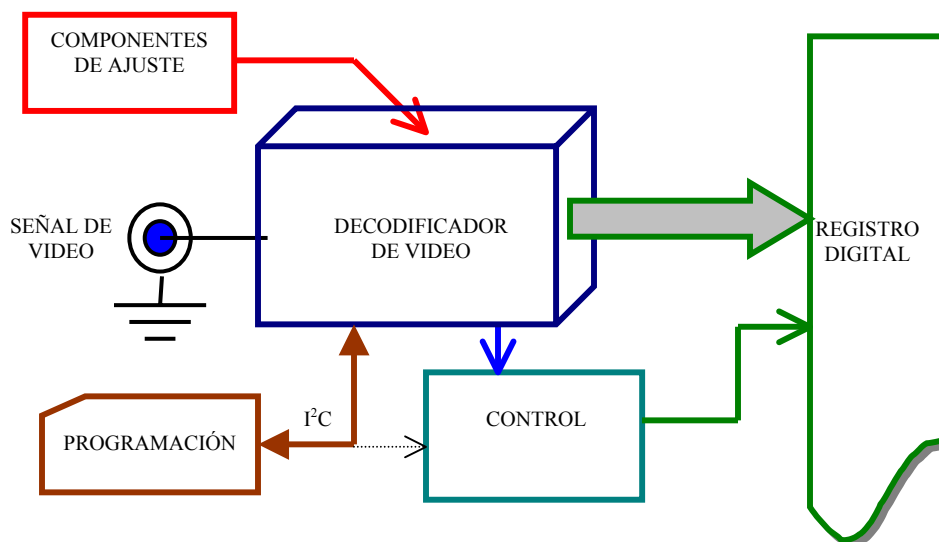


Fig. 4.2 Diagrama de flujo para la digitalización de imágenes de video utilizando un Decodificador de Video.

Dadas las opciones, queda a consideración del diseñador su utilización. Se puede ver que la opción más conveniente es utilizar uno de estos últimos, por lo que en el diseño de este trabajo se usa el decodificador de video S5D0127X01 de Samsung por las ventajas y facilidad de su disponibilidad para la realización de la tesis.

4.1 Planeación

Un sistema orientado a la digitalización de señales de video esta limitado por la cantidad de líneas o renglones que componen la imagen (525 líneas para el caso de NTSC), por lo que la cantidad de píxeles en cada línea debe de ser proporcional al número de éstas para tener una imagen bien proporcionada. En la figura 4.3 observamos diferentes dimensiones de formatos digitales de imágenes con su correspondiente proporción de líneas y píxeles. La proporción de nuestro interés es la del video que se compone de 480 líneas por 640 píxeles para representar una imagen.

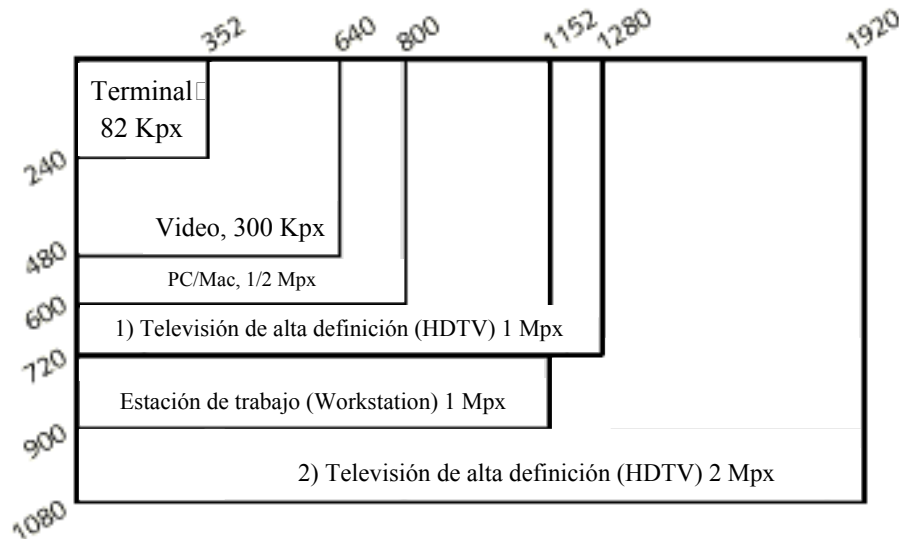


Fig. 4.3 Relación de píxeles y líneas para diferentes dispositivos y formatos estándares de proyección.

Gracias a la utilización del dispositivo S5D0127X01 podemos despreocuparnos por la decodificación y conversión analógico-digital para obtener una imagen y dedicarnos a la recolección de los píxeles para su almacenamiento y a la programación de los registros internos del propio decodificador. Muchos parámetros y condiciones para la digitalización de la imagen se controlan modificando los registros del decodificador como es: Poder cambiar las dimensiones de la imagen hasta un rango máximo de 720 píxeles por línea; cambiar el tamaño del píxel de 16 a 24 bits en colores y contar con todas las señales de sincronización necesarias para identificar las partes que componen la imagen, entre otros parámetros. Con estos parámetros y los datos proporcionados en la ficha técnica del dispositivo que se puede consultar en [[Anexo\1\) Captura de imágenes\S5d0127x01.pdf](#)] se realiza el diseño de acuerdo al esquema de la figura 4.2 y se determinan el material, dispositivos y herramientas necesarios para su construcción.


Nota: De este punto se particulariza para el S5D0127X0, pero existen otros decodificadores de video similares que proporcionan las mismas características y que pueden ser utilizados para sustituir al seleccionado en esta tesis con mínimos ajustes.

4.2 Los dispositivos y componentes

El decodificador de video y dispositivos de ajuste.

Entre los diferentes dispositivos existentes en el mercado se vieron las funciones que realizaban, las capacidades de preprocesamiento, costo, accesibilidad, etc. y de los dispositivos disponibles se pudo obtener por parte del fabricante y su distribuidor los decodificadores de video KS0122 y KS0127 (S5D0127X01), estos solo varían en el tamaño de los datos de salida, voltaje de polarización y funciones extras, como la de teletexto. Entre sus características están:

- Velocidad: tiempo real (30 cuadros por segundo)
- Costo por volumen: ~ 7 dólares.
- Interfaz de configuración accesible; comunicación vía I²C.
- Capacidad de leer diferentes tipos de señales: NTSC, PAL.
- Diferentes tipos de datos de salida RGB, YCrCb 4:2:2, hasta 720 píxeles por línea.
- Frecuencia de trabajo: 24.576 MHz

[ Anexo\1) Captura de imágenes\S5d0127x01.pdf]

La Memoria

Los dispositivos seleccionados y sugeridos para este tipo de decodificador de video son dispositivos de memoria del tipo denominado FIFO (first input, first output FIFO) ‘el primero en entrar, el primero en salir’. Dado la especialización de este tipo de memoria se dispuso de 2 unidades MSM514221B de “OKI semiconductors” proporcionados como muestras, ya que es de difícil obtención en pocas cantidades. Por otra parte sus características son las más convenientes:

- Tamaño de datos: 256 K x 4 bits (2 memorias para un campo (field)).
- Velocidad de lectura o escritura: 25 ns.
- Acceso asíncrono y controlado por solo 3 señales para la entrada: reloj, ‘reset’ y ‘enable’; y 3 para la salida: reloj, ‘reset’ y ‘enable’;

[ Anexo\1) Captura de imágenes\Documentación de dispositivos\MSM514221B.pdf]

Circuito de control

Para mayor flexibilidad se utilizó un dispositivo lógico programable CPLD con capacidad de albergar el control de escritura y lectura de la memoria, la interfaz de comunicación con el decodificador y la interfaz para la transferencia de los píxeles (en el caso de esta tesis se establece un acceso paralelo de dos puertos, uno para el envío de los datos y otro para las palabras de control entre el circuito de captura y la PC).

- Líneas de conexión: 36 pines.
- Velocidad: 5 ns de pin a pin.
- Tamaño: PLCC de 44 pines conteniendo 36 macro celdas con 800 compuertas usables.
- Programación: Con software de Xilinx vía paralelo o serial.

[ Anexo\1) Captura de imágenes\Documentación de dispositivos\CPLDXC9536.pdf].

Los demás componentes utilizados son resistores y capacitores que se emplearon en las etapas de acoplamiento de impedancia, compensación y eliminación de ruido, así como un oscilador para la frecuencia de trabajo del decodificador. Se utilizaron componentes de superficie ya que se acoplan de una mejor forma con el decodificador que es del mismo tipo.

4.3 El diseño

A continuación se presenta una descripción de las partes en que se compone el circuito de captura, utilizando los dispositivos antes mencionados. Todos los archivos de esquemas, diagramas eléctricos y de programación para construir el circuito se encuentran disponibles en el anexo [Anexo\1) Captura de imágenes\Circuito de captura\Modulo del decodificador de video KS0127 y Anexo\1) Captura de imágenes\Circuito de captura \Modulo del control de memoria].

4.3.1 Decodificación de video

El diseño de la etapa de decodificación se centra en poner en marcha el decodificador de video KS0127, para esto se tomó en cuenta: el tipo de señal que se decodifica (en este caso son señales de video compuesto), los voltajes de alimentación y componentes especiales que se necesitaron para su correcto funcionamiento.

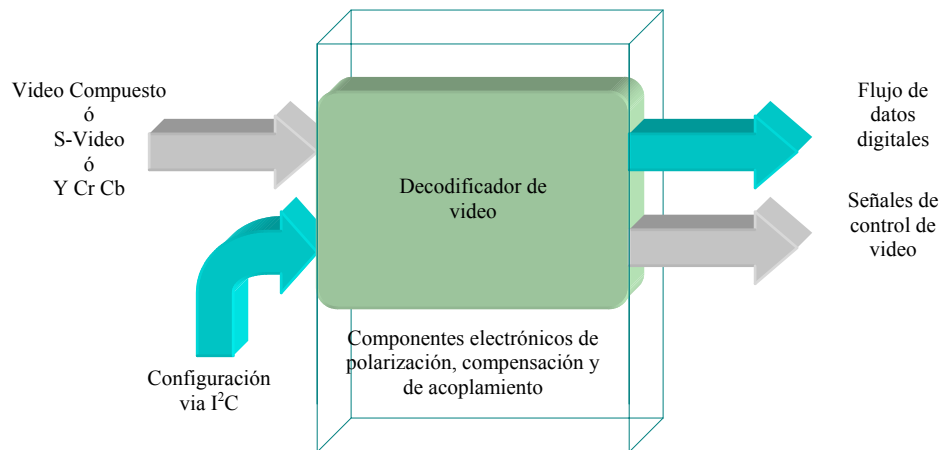


Fig. 4.4 Decodificación de la señal de video

Debido a la peculiar configuración de este tipo de dispositivos se siguieron las especificaciones sugeridas en la ficha técnica, quedando de la siguiente manera: En cada una de las entradas de video se coloca un capacitor cerámico de $0.1\mu\text{F}$ y una resistencia de 75Ω para el acoplamiento de impedancia y aislar al circuito y se colocaron conectores para señales de video compuesto, S-video y YCbCr. Se separaron las vías de polarización para la parte analógica y las digitales de 5 y 3.3 volts respectivamente con componentes denominados 'ferrite-bead' que sirven para evitar la contaminación

de ruido entre ellas. Se le colocó a cada pin de polarización un capacitor de $0.1\mu\text{F}$ de tantalio en forma de 'bypass' para la eliminación de ruido y estabilización del voltaje de alimentación. La configuración del dispositivo se realiza en la etapa de control por medio un puerto I^2C y el número de puerto asignado a éste se complementa con la polarización de los pines 69 y 70 que en este caso están a uno lógico (5 volts). Por último se utilizó un oscilador de 24.576 MHz tipo TTL con un capacitor de $0.1\mu\text{F}$ en 'bypass' en la polarización y la particularidad de necesitar colocar una resistencia (75Ω) hacia el pin de reloj; esto se hizo para estabilizar la amplitud de onda del oscilador con el fin de adecuar esta señal a la requerida por el dispositivo, es importante mencionar que al parecer, este fenómeno se presenta en algunos osciladores con frecuencias altas. En la figura A.1 del apéndice se muestra el esquema de esta parte del diseño.

Una vez polarizado, configurado y conectado a una fuente de video se obtiene a la salida diferentes señales digitales que son de utilidad para capturar los datos (píxeles) de la imagen. Las señales que se generan en forma de pulsos indican: La sincronía horizontal (HS) que genera un pulso positivo al comienzo de cada línea de video. Sincronía vertical (VS) con un nivel de 1 lógico al comienzo de cada campo y dura entre 9 o 10 ciclos de HS. Campo par e impar que es indicado con un nivel de estado lógico 0 o 1 respectivamente (ODD). Video activo (AV) que está en 1 lógico cuando hay datos válidos. Un pulso de reloj (CK2) que señala cuando hay un píxel válido y también es el que lleva el ritmo de la generación de estos. Estas señales son suficientes para poder almacenar la imagen en memoria.

Los píxeles son puestos a la salida en sincronía con CK2 en dos buses de señales 'C₇-C₀' y 'Y₇-Y₀' a un ritmo de 30 cuadros por segundo, con una resolución de 720 píxeles por línea y con una cantidad de 16 bits en color u 8 bits en blanco y negro que representan 256 niveles de gris.

4.3.2 Memoria de datos

La memoria es un componente indispensable para la captura de la imagen, dependiendo de la cantidad de memoria con que se disponga puede almacenarse un campo, un cuadro, solo en blanco y negro, a colores o toda una secuencia de imágenes. Otro factor que determina la cantidad de memoria es el número de muestras por línea, aunque este factor debe ser proporcional al número total de líneas que es constante (525 para un cuadro completo y 252.5 para un solo campo en NTSC). Si se captura una imagen en blanco y negro con un tipo de dato entero de 8 bits y muestras por línea de 720 y 320 en video activo (AV) de 488 y 244 líneas respectivamente tienen la siguiente relación:

Imagen	Píxeles por líneas	Memoria en KB
Cuadro completo	720 x 488	344
Campo par o impar	720 x 244	172
Campo par o impar proporcional	360 x 244	86

Tabla 4.1 Espacio de las Imágenes en Kilo Bytes.

Para el caso de la tesis se decidió por una imagen de 360 píxeles por 244 líneas en blanco y negro que tiene un tamaño y resolución aceptable para muchas aplicaciones y por que los módulos de memoria ‘FIFO’ MSM514221B con que se cuentan son solo suficientes para un campo ya sea par o impar con 256 niveles de gris. La insistencia en el uso de este tipo de memoria esta motivada por lograr un incremento de velocidad en escritura y lectura, así como una previsión para las mejoras a futuro ya que este tipo de memoria es relativamente barato pero difícil de conseguir en pocas cantidades (Las características de la memoria eléctricas como de tiempos se pueden consultar en el anexo [Anexo I Captura de imágenes\Documentación de dispositivos\MSM514221B.pdf]). Las señales necesarias para la escritura y lectura se generan en la etapa de control utilizando a su vez las señales ‘ODD’, ‘AV’, ‘HS’, ‘CK2’ y ‘Y₇-Y₀’ del decodificador de video (Y₇-Y₀ componen los píxeles de la imagen en escala de grises).

La operación de este tipo de memoria resulta muy sencillo, esta se maneja por dos grupos de señales: WE, RSTW y SWCLK para la escritura y RE, RSTR y SRCLK para la lectura y solo requiere de un banco de capacitores ‘bypass’ por cada cos componentes. La figura A.2 del apéndice muestra el esquema utilizado en la construcción del circuito.

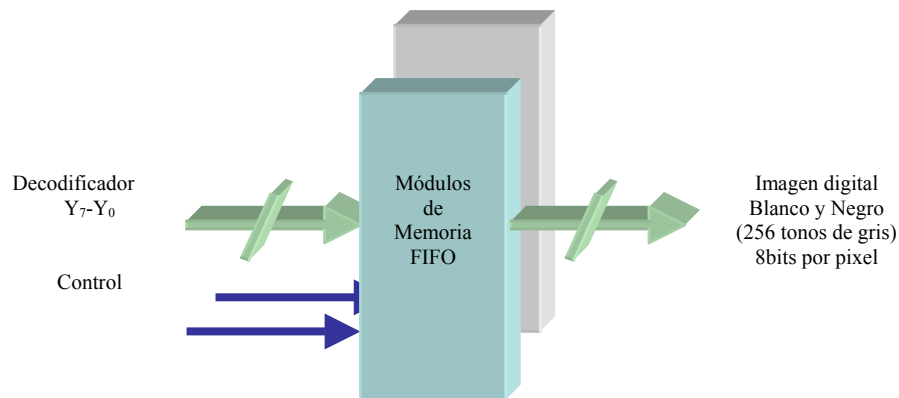


Fig. 4.5 Almacenamiento de la imagen en memoria.

Para capturar la imagen se realiza lo siguiente: cuando un campo de imagen comienza se genera una señal de ‘reset’ en RSTW de a la memoria para que apunte al inicio de los registros, por cada pulso de reloj en SWCLK es introducido un píxel siempre que la señal WE es activa (1 lógico), cuando es desactivada (cero lógico) la escritura se detiene; de manera similar se realiza la lectura utilizando RSTR para apuntar al inicio de los registros y sacando un píxel por cada ciclo de reloj en SRCLK sólo si la señal RE esta activa (1 lógico) y al desactivarla, la salida pasa a tercer estado deteniendo el flujo de los datos. Las tareas de refresco de memoria se realizan de manera automática.

4.3.3 Lógica de control

La etapa de control es la más importante ya que con esta se maneja la escritura y lectura de la imagen en la memoria FIFO, se encarga de acoplar el bus de comunicación I²C del decodificador para

que pueda conectarse a un puerto paralelo. Por último se implementa la lógica de control para comunicar el circuito de captura con la computadora personal (PC).

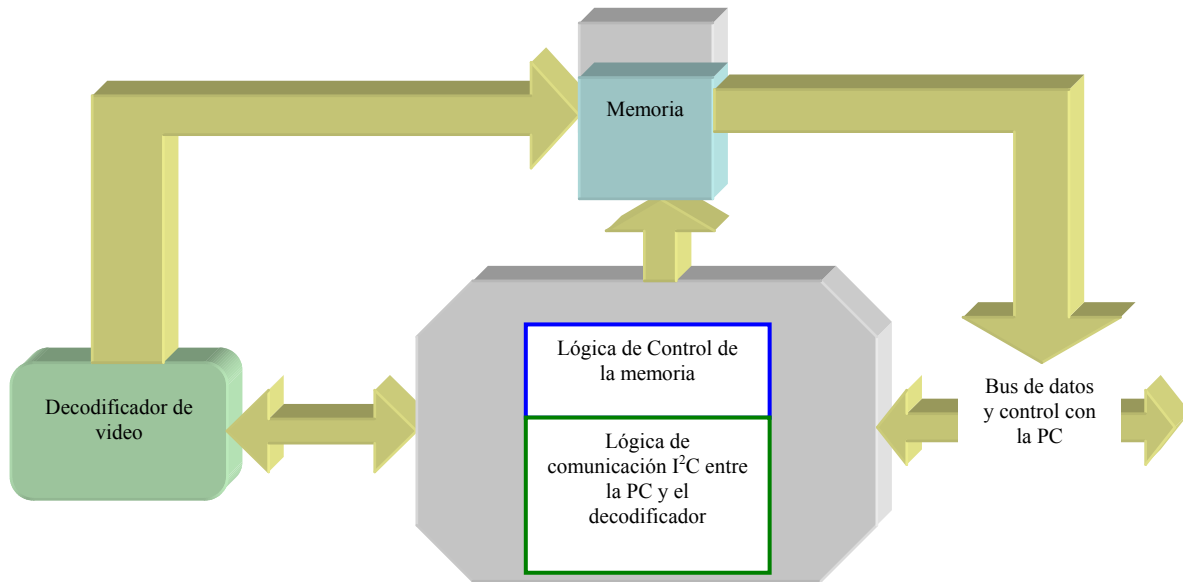


Fig. 4.6 Diagrama de la lógica de control.

Para facilitar el diseño y aprovechar otros trabajos realizados se utilizó una interfaz paralela con la PC de dos puertos (esta es una de las partes que se deja como mejoras a futuro que se ve en la sección 7.3 del capítulo de resultados y conclusiones). El funcionamiento es sencillo, se tiene una tarjeta que permite enviar y recibir datos por los puertos 0x0303 y 0x030C mediante instrucciones ejecutadas desde la PC mediante un programa implementado en lenguaje C. El bus de conexión es de tipo DB25 y se conforma de un los datos (8 bits: D₇-D₀ Pin8 – Pin1), señales de voltajes para la alimentación del sistema (5v Pin25, -5v Pin13, 12v Pin24, -12v Pin12 y GND Pin11, Pin14 – Pin21) así como de las señales de -IOW (Pin23) y -IOR (Pin10) para la escritura y lectura respectivamente y las señales -303 (Pin9) y -30C (Pin22) que son implementadas en un circuito lógico programable (La lógica es la condición AND que se presenta para 0x0303 y 0x030C en el bus de direcciones A₁₉-A₀ de la PC). Como nota de diseño se mencionan que se colocaron capacitores 'bypass' en las señales de voltaje para compensar la longitud del cable utilizado (en cualquier caso se deben de respetar las mismas reglas de longitud que se aplican a un cable para impresora).

La escritura y lectura de los píxeles en la memoria FIFO se realiza mediante lógica combinatoria controlada desde la PC, ya que en esta es donde finalmente es almacenada la imagen capturada. El proceso de escritura en memoria se hace utilizando las señales ODD, AV, HS, VS y CK2 de la siguiente forma: Se tiene un flujo de escritura permanente que es llenada con la información del campo impar, cada vez que éste se encuentra presente (ODD en uno lógico) y se genera el pulso de sincronía vertical (VS) se genera un pulso en la señal RSTW en los módulos de memoria para volver a llenarlos desde el registro 0, los píxeles son almacenados al ritmo de la señal CK2 en SWCLK solo y solo si existe información válida (WE = AV & /HS). Durante el proceso de lectura se detiene la escritura sólo con el fin de no alterar la información capturada, mientras esto es así, se procede a realizar un ciclo de lecturas continuas mandando la siguiente secuencia: un pulso en RSTR para

apuntar al inicio de la información, se habilita la salida de información (RE) y se lee cada píxel dando un pulso en la señal SRCLK, al finalizar la lectura se vuelve a reanudar el proceso de escritura permanente.

Para la comunicación en I²C se implementa una lógica digital que permite realizar el acoplamiento del bus del decodificador de video y la interfaz paralela mediante un simulador de este protocolo (el I²C es un estándar de comunicación en diferentes dispositivos dedicados a la electrónica de consumo, para ver todo lo relacionado con este protocolo, su implementación y el modelo del simulador puede consultarse en el anexo [**Anexo\1) Captura de imágenes\I2C**]). La configuración del decodificador de video se realiza desde la PC en donde se lee por un puerto la información que viene en forma serial y es convertida en paralelo para su interpretación, de forma contraria los datos y palabras de control son enviadas al dispositivo reconvirtiendo de paralelo a serial y empaquetándolos en el formato del protocolo I²C.

Todos los datos y parámetros que se pueden modificar en el decodificador de video así como las reglas que se deben de seguir al hacer algún cambio se encuentran descritos la ficha técnica del propio dispositivo [**Anexo\1) Captura de imágenes\S5d0127x01.pdf**]. En general entre las cosas que se pueden modificar están: La frecuencia de corte del filtro para la separación de luminancia y croma, seleccionar entre diferentes algoritmos de separación, determinar los niveles de brillo, contraste y tinte de la imagen, seleccionar la señal de entrada de 6 diferentes posibilidades así como el tipo y su calidad, controlar el consumo de energía, modificar el muestreo de los datos de salida y determinar el formato de los mismos, etc.

Toda la lógica de control o circuito digital utilizado para manejar las señales de entrada y salida de las memorias, la comunicación para configurar al decodificador y encargarse de establecer la interfaz de conexión con la PC se implementó en unos dispositivos lógico programables CPLD XC9536 [**Anexo\1) Captura de imágenes\Documentación de dispositivos\CPLDXC9536.pdf**] por su gran versatilidad, reducción de líneas y espacio de circuito impreso; y un buffer TTL 74LS245. En la figura A.3 del apéndice se muestra el diagrama esquemático de la etapa de control.

La programación del CPLD se realizo con la herramienta de diseño ‘Foundation’ de ‘Xilinx’ utilizando para ello el diseñador esquemático donde se interconectaron los módulos transferencia de datos, lectura y escritura a memoria y el protocolo de comunicación I²C para la configuración desde la computadora. Las figuras A.4, A.5 y A.6 del apéndice muestran los diagramas lógicos que se utilizaron para la programación.

4.3.4 Circuito impreso

Con la ayuda del programa ‘OrCAD Ver.9’ se diseñó y terminó el modelo esquemático del circuito de captura y se procedió a realizar el circuito impreso. El proceso de enrutamiento del circuito impreso se llevó a cabo una gran parte del tiempo ya que tuvo que implementarse en un modelo de 2 capas sabiendo que los dispositivos utilizados están orientados a un diseño multicapa y deben estar lo más cercano posible al decodificador. Una vez terminado este paso se tiene un diseño que es depurado y convertido a un formato para la elaboración física (Esta etapa se realizó mediante los programas de LPKF: CircuitCAM 3.0 y BoardMaster 3.0). Al final se procedió a fabricar en la máquina ‘ProtoMat 95s/II de LPKF’ el circuito impreso en una tablilla de cobre. En las figuras A.7 y A.8 del apéndice se muestran las imágenes de los circuitos impresos elaborados en esta tesis.

CAPÍTULO 5: PREPROCESAMIENTO DE IMÁGENES

5

Como esta parte de la tesis se realiza utilizando un sistema de desarrollo ya construido, se toma la decisión de realizar una implementación paralela entre un sistema basado en un paquete de desarrollo con procesador de señales digital (DSP) y como un modulo adicional, otro que se implementa como una biblioteca en el programa de visualización que reside en la propia computadora como sistema de preprocesamiento alterno. La razón es que al utilizar la computadora para las otras dos partes de la tesis y considerando a ésta como un sistema de hardware más, es obvio que puede desempeñar el papel del sistema de preprocesamiento. Además que nos permitió enfocarnos más en los algoritmos que se implementaron y aplicaron a las imágenes y observando las ventajas y desventajas que se dieron al utilizar cada uno de los sistemas. La forma en que interactuaron cada uno de los dos sistemas fue en el sentido de mandar los datos de la imagen original y recibir el resultado para ser evaluados, como se muestra en el esquema de la figura siguiente.



Fig. 5.1 Diagrama de flujo del sistema de preprocesamiento de imágenes digitales.

La etapa de digitalización captura las imágenes almacenándolas en archivos y el sistema de visualización las lee para su representación en pantalla. La etapa de preprocesamiento lee los archivos que conforman los datos la imagen, aplica el algoritmo y genera un resultado que es colocado en un nuevo archivo para su visualización o almacenamiento. Veremos a continuación los algoritmos, su implementación y la adecuación de los datos; finalmente se analiza el desempeño de los sistemas mencionados.

5.1 Implementación

Además de la descripción del o de los algoritmos, se describe el método utilizado para aplicarlo a las imágenes capturadas.

Del modelo matemático descrito en el capítulo 2 sección 3 y del diseño resultante del circuito de captura, los datos de la imagen confirman una matriz $f(x,y)$ de 244 renglones en el eje Y, por 360 columnas en X. El origen se considerara de acuerdo al algoritmo, pero por lo general se toma el lado superior izquierdo que se basa en la forma en que es capturada la imagen. Los valores vienen ordenados de izquierda a derecha y de arriba a bajo, igual que en la secuencia de barrido de un televisor.

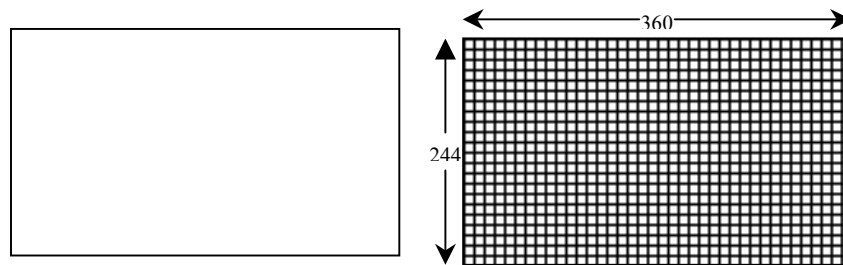


Fig. 5.2 Conjunto ordenado que conforma los datos en la imagen.

El tipo de dato a manejar es el descrito en la solución propuesta sección 4.3 que define a una imagen en blanco y negro con valores enteros de 8 bits que van de 0 para el negro hasta el 255 para el blanco. Los parámetros necesarios para cada algoritmo están dados en el mismo tipo de datos y en números flotantes denominados como r_i , el resultado final siempre será entero y definido en una matriz $\mathfrak{g}(x,y)$.

Los pasos que se realizan para la aplicación de los algoritmos se describen en el esquema de la figura 5.3 de la siguiente forma: una o dos imágenes originales son leídas de sus respectivos archivos o tomada una de ellas directamente del circuito de captura y enviadas al sistema de preprocesamiento (si es el caso se añaden parámetros o variables adicionales), una vez que los datos han entrado se aplica el algoritmo que fue seleccionado y previamente programado. Al final los datos resultantes son sacados y enviados para su almacenamiento en un nuevo archivo o presentados en pantalla.

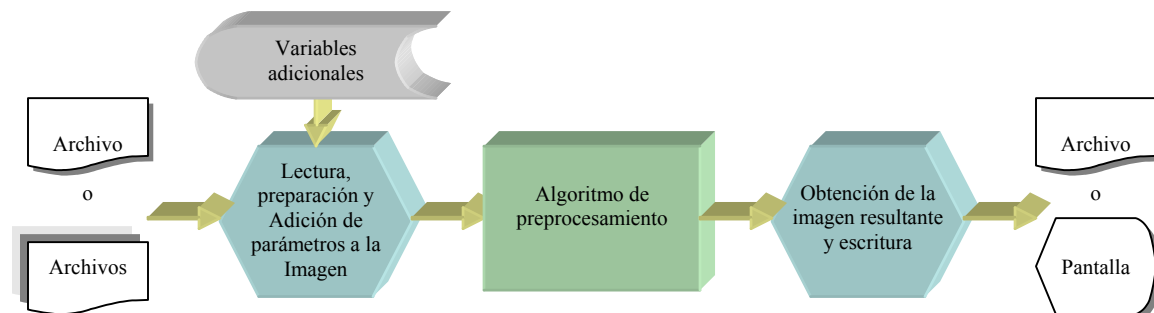


Fig. 5.3 Pasos para la aplicación del algoritmo de preprocesamiento a las imágenes.

Para la implementación en el sistema basado en el procesador de señales digital (DSP) se utilizó el programa 'Code Composer Studio (CCS)' que genera los vectores de interrupciones, el mapeo de memoria, los temporizadores, etc. y demás periféricos para construir una aplicación completa y que pueda ser depositada en el sistema a través de su interfaz con la PC. Por otro lado un sistema de preprocesamiento basado en la PC no es más que un programa o biblioteca de funciones que pueden ser creadas por un compilador que maneje cierto lenguaje, como es 'C++' en este caso. La forma en que se trabajó con cada sistema fue algo diferente ya que se tenían ambientes diferentes, esto se describe más adelante; pero antes mencionamos los algoritmos que se implementaron en ambos sistemas.

5.2 Algoritmos

De las diversas funciones mencionadas en el estado del arte (sección 2.4) se implementaron algunas que representan la diversidad de éstas, sin dejar en cuenta que existen muchas más que pueden ser implementadas. Las funciones están diseñadas para el tamaño (360 * 244) y tipo de dato (entero sin signo) que entrega el circuito de captura.

Funciones básicas y lógicas

De estas se implementaron las más representativas, mostrando la parte en donde radica la diferencia de cada una de ellas. Dadas dos imágenes $f(x,y)$ y $f'(x,y)$, donde $x_i \in [0,243]$ y $y_j \in [0,359]$ las ecuaciones que determina cada operación y su algoritmo son las siguientes:

Suma, Resta, Máximo, Mínimo, And, Or, Xor: $\mathcal{G}(x_i, y_j) = f(x_i, y_j)(+, -, >, <, \&, |, ^) f'(x_i, y_j)$

Not: $\mathcal{G}(x_i, y_j) = 255 - f(x_i, y_j)$

Para j = 0 hasta 243 hacer

Para i = 0 hasta 359 hacer

Suma, Resta, Máximo, Mínimo, And, Or, Xor, Not //parte en donde se modifica la función

Termina i = 359

Termina j = 243

Umbralizado

Aunque es muy sencilla, esta es de gran utilidad en diversas áreas como la segmentación y el reconocimiento de patrones. Sea $f(x,y)$ la imagen y $r \subseteq [0-255]$ el parámetro de umbralizado, la ecuación es:

Umbralizado $\mathcal{G}(x_i, y_j) = \{ \{ 0 \Leftrightarrow f(x_i, y_j) < r \}, \{ 255 \Leftrightarrow f(x_i, y_j) \geq r \} \}$

Para j = 0 hasta 243 hacer

Para i = 0 hasta 359 hacer

Si $f(x_i, y_j)$ mayor o igual a 'r' $\mathcal{G}(x_i, y_j) = 255$

Otro $\mathcal{G}(x_i, y_j) = 0$

Termina i = 359

Termina j = 243

Geométrica

Para la representación de las funciones de este tipo se implementó la de “Compensación de lente” que es útil cuando la imagen proviene de una cámara de video con un lente denominado “ojo de pescado” usada regularmente en seguridad, la cual abarca un área esférica en vez de una plana; lo que se tiene aquí es una transformación de coordenadas, ya que la imagen a preprocesar es una vista plana de la superficie de una esfera. Y por lo tanto esa superficie esférica debe ser trasladada de 3 a 2 dimensiones.

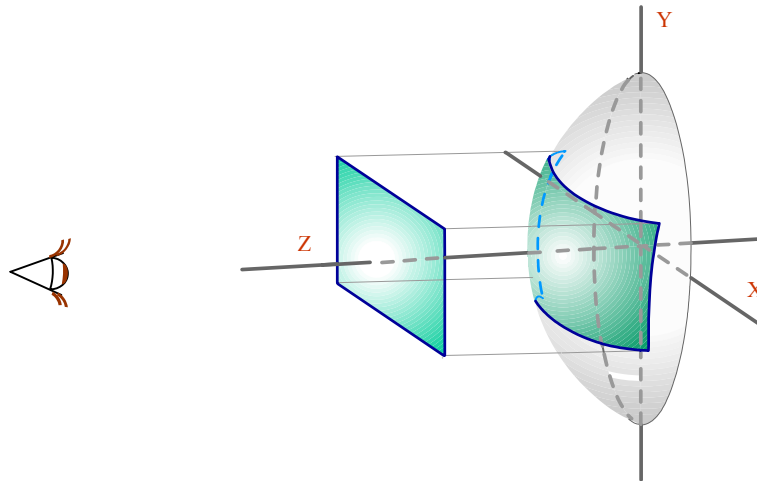


Fig. 5.4 Perspectiva plana de una imagen en espacio esférico.

Los datos necesarios para la transformación son: el ángulo ϕ en grados de la lente y que cubre el área que abarca la imagen (este valor puede ser mayor de cero y hasta 180 grados) y el área que abarca la vista plana de la imagen (como unidades se toma la cantidad de píxeles, siendo 244 para el eje Y y 360 en X) tomando como origen el centro de esta; y aunque el número de grados es proporcionado por el fabricante de la lente o la cámara se debe de considerar una cierta tolerancia por pequeñas variaciones. Ya que para los ejes X y Y se establece la misma relación respecto al eje Z y la imagen, se hace el análisis para X-Z y se transponla a Y-Z.

Sean x,y las coordenadas del plano sobre la imagen en formato esférico y vistas desde el eje z, frontal a la lente y $Nx=360$, $Ny=244$ el número de muestras en X y Y respectivamente. Para obtener el radio γ de la esfera que tiene en su superficie la imagen realizamos lo siguiente:

El área del rectángulo esta delimitada por Nx y Ny , también se establece a las esquinas del área como los limites del ángulo ϕ que abarca la imagen y donde el radio γ es el mismo. Se tiene entonces que del centro a cualquier esquina el ángulo es $\phi/2$, por otro lado la distancia del mismo centro hacia cualquier esquina en el plano esta dada por $\sqrt{Nx^2 + Ny^2} / 2$ y que es en si el cateto opuesto del radio que se esta buscando. La ecuación $\text{sen}(\Phi) = \frac{\text{cateto_opuesto}}{\text{hipotenusa}}$ se utiliza para obtener el radio.

$$\text{hipotenusa} = \frac{\text{cateto_opuesto}}{\text{sen}(\Phi)} \rightarrow \gamma = \frac{\sqrt{Nx^2 + Ny^2}}{2 \text{sen}\left(\frac{\phi}{2}\right)} \quad (\text{eq. 5.1})$$

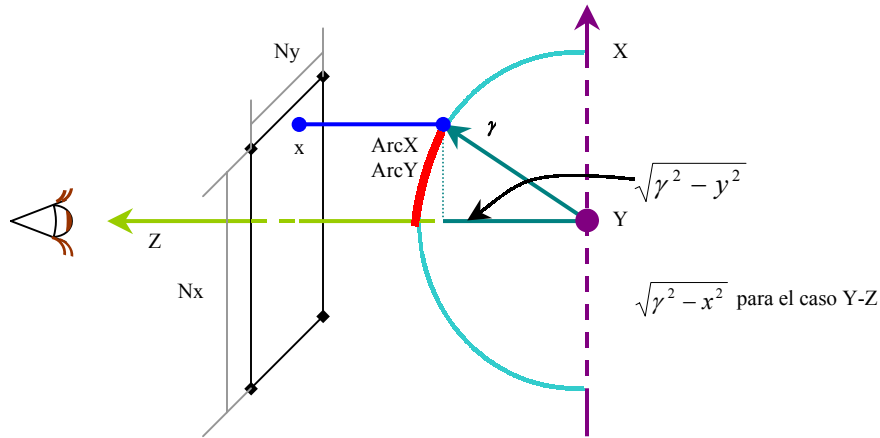


Fig. 5.5 Esquema para determinar las distancias ArcX y ArcY.

Finalmente como la imagen debe de ser colocada nuevamente en un plano tenemos entonces que encontrar la nueva posición en el eje **X**. Hay que determinar el arco-distancia desde el centro y para esto utilizamos $X_{arc} = \gamma * \text{Arco-sin}(\omega)$, donde $\omega = \frac{x}{\sqrt{\gamma^2 - y^2}}$ ya que el ángulo utilizado para sacar el arco-distancia es el que se encuentra sobre la circunferencia en el plano X-Z con centro en 'y' del eje Y. Con un radio máximo igual a z y γ siempre mayor a y o x tenemos:

$$X_{arc} = \gamma \text{Asen} \left(\frac{x}{\sqrt{\gamma^2 - y^2}} \right) \tag{eq. 5.2}$$

haciendo lo mismo para el eje **Y** la ecuación quedan como:

$$Y_{arc} = \gamma \text{Asen} \left(\frac{y}{\sqrt{\gamma^2 - x^2}} \right) \tag{eq. 5.3}$$

y el algoritmo es:

Para $j = -122$ hasta 122 hacer
 Para $i = -180$ hasta 180 hacer
 $\mathcal{G}(X_{arc}, Y_{arc}) = f(x, y);$
 Termina $i = 180$
 Termina $j = 122$

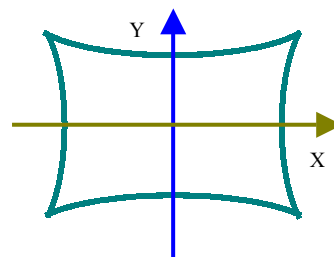


Fig. 5.6 Forma de la imagen final al trasladarse de una superficie esférica a una plana.

Convolución y correlación

Para esto se implementó el método espacial descrito en la sección 2.3 y 2.4.6 donde se utiliza una ventana (matriz) μ de 3x3 y otra de 5x5 con pesos (valores) que pueden ser modificados y con un valor κ de compensación para adecuar el resultado al tipo de dato utilizado.

$$\mu = \begin{bmatrix} m11 & m12 & m13 \\ m21 & m22 & m23 \\ m31 & m32 & m33 \end{bmatrix} * \kappa \quad (\text{eq. 5.4})$$

ó

$$\mu = \begin{bmatrix} n11 & n12 & n13 & n14 & n15 \\ n21 & n22 & n23 & n24 & n25 \\ n31 & n32 & n33 & n34 & n35 \\ n41 & n42 & n43 & n44 & n45 \\ n51 & n52 & n53 & n54 & n55 \end{bmatrix} * \kappa \quad (\text{eq. 5.4})$$

EL procedimiento es el siguiente: se aplica sobre cada píxel de la imagen y donde no existe un valor se determina a éste igual a cero, de esta forma el algoritmo queda como sigue:

Para $j=0$ hasta 243 hacer

Para $i=0$ hasta 359 hacer

$$\mathcal{G}(x_i, y_j) = (f(x_{i-1}, y_{j-1}) * m11 + f(x_i, y_{j-1}) * m12 + f(x_{i+1}, y_{j-1}) * m13 + \\ f(x_{i-1}, y_j) * m21 + f(x_i, y_j) * m22 + f(x_{i+1}, y_j) * m23 + \\ f(x_{i-1}, y_{j+1}) * m31 + f(x_i, y_{j+1}) * m32 + f(x_{i+1}, y_{j+1}) * m33) / \kappa$$

Termina $i=359$

Termina $j=243$

ó

Para $j=0$ hasta 243 hacer

Para $i=0$ hasta 359 hacer

$$\mathcal{G}(x_i, y_j) = (\\ f(x_{i-2}, y_{j-2}) * n11 + f(x_{i-1}, y_{j-2}) * n11 + f(x_i, y_{j-2}) * n12 + f(x_{i+1}, y_{j-2}) * n13 + f(x_{i+1}, y_{j-2}) * n13 + \\ f(x_{i-2}, y_{j-1}) * n11 + f(x_{i-1}, y_{j-1}) * n11 + f(x_i, y_{j-1}) * n12 + f(x_{i+1}, y_{j-1}) * n13 + f(x_{i+1}, y_{j-1}) * n13 + \\ f(x_{i-2}, y_j) * n21 + f(x_{i-1}, y_j) * n21 + f(x_i, y_j) * n22 + f(x_{i+1}, y_j) * n23 + f(x_{i+1}, y_j) * n23 + \\ f(x_{i-2}, y_{j+1}) * n31 + f(x_{i-1}, y_{j+1}) * n31 + f(x_i, y_{j+1}) * n32 + f(x_{i+1}, y_{j+1}) * n33 + f(x_{i+1}, y_{j+1}) * n33 + \\ f(x_{i-2}, y_{j+2}) * n31 + f(x_{i-1}, y_{j+2}) * n31 + f(x_i, y_{j+2}) * n32 + f(x_{i+1}, y_{j+2}) * n33 + f(x_{i+1}, y_{j+2}) * n33 \\) / \kappa$$

Termina $i=359$

Termina $j=243$

5.3 Adecuación de resultados

Al preprocesar una imagen con un determinado algoritmo se llega a dar el caso en que el resultado obtenido no puede registrarse adecuadamente en la nueva imagen; esto sucede por ejemplo, cuando la imagen es alterada por una función geométrica en donde la posición del píxel cambia o cuando el valor del nuevo píxel llega a ser mayor o inferior al rango de valores que se están manejado. Para esto se deben de considerar estrategias que manejen estas particularidades.

En el caso en que la posición del nuevo píxel es diferente a la original ya sea por una diferencia entera o fraccionaria se ha implementado el siguiente algoritmo:

Sean X, Y los valores enteros o entero-fraccionarios de la nueva posición del píxel resultantes del hecho de haber aplicado una función geométrica a x, y (ej: $X = 3.57, Y = 7.01$), entonces tenemos que $X_{int}, Y_{int}, X_{frac}, Y_{frac}$ son las porciones entera y decimal de X , (ej: $X_{int} = 3, Y_{int} = 7, X_{frac} = 0.57, Y_{frac} = 0.01$). Para la colocación del píxel en su nueva posición se tomó el siguiente convenio: El píxel va a ser colocado sobre la posición de mayor coincidencia, y para esto se considera un espacio de 4 posiciones en el entorno de X, Y donde $X \in [X_{int}, X_{int} + 1]$ y $Y \in [Y_{int}, Y_{int} + 1]$. Además se consideran umbrales de coincidencia para cada coordenada se asignan como sigue: Si la parte fraccionaria de la coordenada esta entre 0 y 0.33 la posición que le corresponde es la de su valor entero (ej: $X = X_{int}$), si la parte fraccionaria esta entre 0.66 y 0.99 la nueva posición será la del valor entero más uno (ej: $X = X_{int} + 1$) y finalmente en el caso especial en que la posición se encuentre entre 0.34 y 0.65 se ocupara tanto el valor entero y el siguiente (ej: $X = X_{int}$ y $X = X_{int} + 1$) con la observación de que si en la posición donde se va a colocar el píxel ya existe otro se tiene que realizar el promedio de ambos.

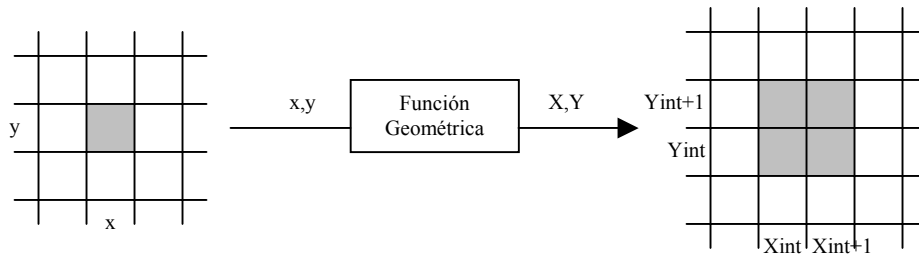


Fig. 5.7 Manejo de la posición del píxel cuando sus valores no son enteros absolutos.

Tanto para X como para Y :

Si $X_{frac} < 0.33$ y $Y_{frac} < 0.33$ poner píxel en X_{int}, Y_{int}

Si $X_{frac} < 0.33$ y $0.33 \leq Y_{frac} \leq 0.66$ poner píxel en X_{int}, Y_{int} y $X_{int}, Y_{int}+1$

Si $X_{frac} < 0.33$ y $0.66 < Y_{frac}$ poner píxel en $X_{int}, Y_{int}+1$

Si $0.33 \leq X_{frac} \leq 0.66$ y $Y_{frac} < 0.33$ poner píxel en X_{int}, Y_{int} y $X_{int}+1, Y_{int}$

Si $0.33 \leq X_{frac} \leq 0.66$ y $0.33 \leq Y_{frac} < 0.66$ poner píxel en X_{int}, Y_{int} $X_{int}, Y_{int}+1$ $X_{int}+1, Y_{int}$ y $X_{int}+1, Y_{int}+1$

Si $0.33 \leq X_{frac} \leq 0.66$ y $0.66 < Y_{frac}$ poner píxel en $X_{int}, Y_{int}+1$ y $X_{int}+1, Y_{int}+1$

Si $0.66 < X_{frac}$ y $Y_{frac} < 0.33$ poner píxel en $X_{int}+1, Y_{int}$

Si $0.66 < X_{frac}$ y $0.33 \leq Y_{frac} \leq 0.66$ poner píxel en $X_{int}+1, Y_{int}$ y $X_{int}+1, Y_{int}+1$

Si $0.66 < X_{frac}$ y $0.66 < Y_{frac}$ poner píxel en $X_{int}+1, Y_{int}+1$

En cualquiera de los casos, si en la nueva posición P_x, P_y ya existe un valor de píxel entonces:

Nuevo píxel(NP_x, NP_y) = (píxel(X, Y) + píxel(P_x, P_y)) / 2

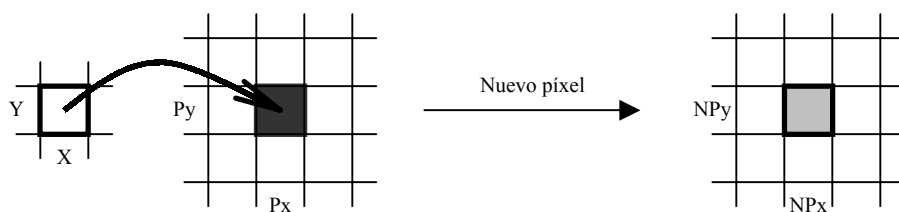


Fig. 5.8 Combinación de valores de píxel en la misma posición.

Cuando el valor resultante de una función es mayor o menor del conjunto de valores permitidos para un píxel o no pertenece a alguno de ellos, se tiene que realizar un ajuste para que encaje dentro. Lo más conveniente es el acotamiento y redondeo. El acotamiento permite cortar o limitar el valor resultante para que ajuste dentro del conjunto, esto se hace al asignar el valor límite correspondiente. Si consideramos al conjunto de valores permitidos dentro del rango $[0, L]$ (ver eq. 2.4) todo valor inferior a 0 pasará ser éste y de la misma forma todos aquellos mayores a L se les asignará este último.

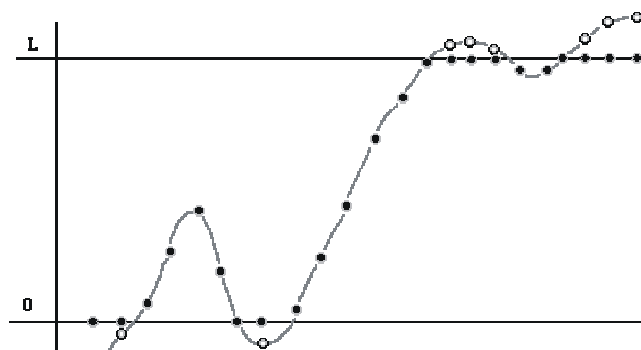


Fig. 5.9 Acotamiento de los valores de un píxel.

Otro ajuste que se tiene que realizar es el de tipo de dato. Si se está manejando por ejemplo un tipo de dato entero, y el resultado de alguna función es fraccionario se debe de realizar un redondeo, truncarlo o seleccionar el valor entero inferior o superior; de estas opciones, el truncado y el redondeado son dos técnicas fáciles de implementarlas de manera práctica.

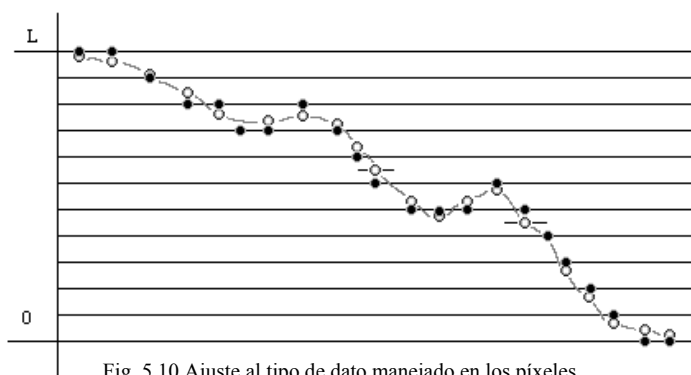


Fig. 5.10 Ajuste al tipo de dato manejado en los píxeles.

5.4 Sistemas de preprocesamiento y componentes

Para la implementación de los algoritmos se cuenta con 2 posibilidades. Si ya se tiene una computadora como receptora de las imágenes capturadas es común suponer el uso de ésta para realizar el preprocesamiento, sin embargo si se quiere realizar una aplicación dedicada es conveniente utilizar un sistema independiente. Por tal razón se ha decidido hacer la implementación en ambos sistemas para que se puedan comparar y evaluar, entendiendo esto como sistema de modelado en donde se puede realizar ajustes y pruebas y otro donde se puedan implementar aplicaciones dedicadas para algún fin.

El sistema de preprocesamiento debe de tener ciertas características que son necesarias para el manejo de imágenes. Esencialmente se debe de contar con una unidad aritmética lógica potente para poder implementar algoritmos que muchas veces se componen de operaciones matemáticas complejas que consumen muchos ciclos de procesamiento, el otro requerimiento es contar con el espacio de memoria suficiente para poder manejar y almacenar temporalmente todos los datos que componen una imagen y en algunos casos otros datos más.

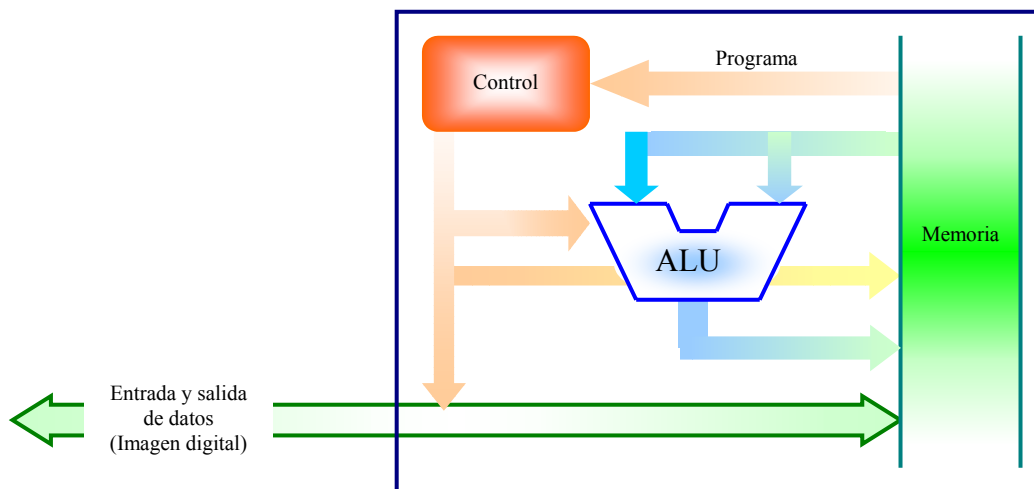




Fig. 5.11 Componentes básicos de un sistema de preprocesamiento.

Para el caso de esta tesis se tiene por un lado la computadora personal que en la actualidad cuenta con una unidad aritmética y lógica que permite realizar complejas operaciones y algoritmos de preprocesamiento de imágenes, la ventaja de la velocidad de procesamiento con que se cuentan ahora y la cantidad de memoria de trabajo real y virtual que puede generar por medio del sistema operativo es una alternativa como sistema de preprocesamiento. La otra opción que se presenta es la de un sistema enfocado a aplicaciones específicas y de gran velocidad de procesamiento, éste es un sistema de desarrollo con un procesador de señales digital con memoria de trabajo interna y externa que proporciona un ambiente de trabajo para el desarrollo de algoritmos en lenguaje de alto nivel (C++), esta herramienta es útil para la implementación y prueba en aplicaciones específicas y dedicadas.

5.4.1 La computadora personal como sistema de preprocesamiento

La utilización de la computadora personal como sistema de preprocesamiento es un extra en el diseño del sistema propuesto en esta tesis, ya que su implementación es una consecuencia del programa de visualización que se ve en el capítulo siguiente. Solo queda mencionar que esta parte sirve como ayuda en la generación, construcción y depuración de los algoritmos que después se implementan en un sistema independiente como el que se utiliza es en este caso, basado en el procesador digital de señales.

La etapa de preprocesamiento realizado en la PC se diseñó como una biblioteca de funciones donde se introducen las imágenes como arreglos bidimensionales de valores enteros con otras variables si es el caso y se regresa el resultado en el mismo u otro arreglo. El espacio de trabajo se definió como un arreglo 2 veces más grande del tamaño de la imagen para poder trabajar con ellas de una manera fácil y se agregaron funciones especiales para su limpieza y preparación. Como se programó en lenguaje C++ con un compilador de alto nivel orientado a objetos esta biblioteca de funciones se encuentra dentro de un objeto denominado 'Proc' [ Anexo\3) Programa de visualización\Programa PROIMG\proc.h y proc.cpp] y que puede ser exportable y acoplado a otros programas. La manera en que opera este objeto es mostrada en la figura 5.12 y puede ser consultado en los archivos del programa en [ Anexo\3) Programa de visualización\Programa PROIMG\preproc.cpp].

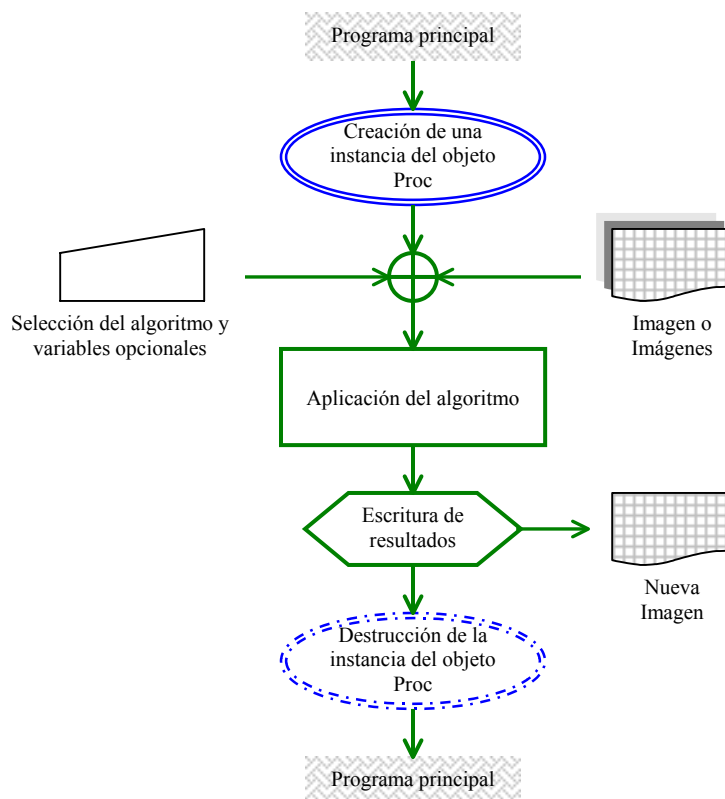


Fig. 5.12 Operación del modulo de preprocesamiento en la PC.

Aunque esta parte se entiende como un trabajo de software y no de hardware tiene merito propio.

5.4.2 Sistema de preprocesamiento basado en un procesador de señales digital

Entre muchas razones para utilizar un procesador de señales digital y en este caso un sistema de desarrollo ya construido es por que se deja abierta la posibilidad de que una vez implementado, depurado y probado un algoritmo en una aplicación específica puede entonces decidirse a construir un prototipo que incluya la captura y el preprocesamiento de imágenes para una determinada aplicación.

La característica principal del sistema utilizado es que proporciona un ambiente de trabajo que nos permite generar a partir de una programación en lenguaje de alto nivel archivos binarios que además de contener el algoritmo de preprocesamiento, proporciona las instrucciones y configuraciones para el manejo de interrupciones, temporizadores y demás periféricos internos del procesador de señales digital que se necesiten en un proyecto específico. La desventaja de este sistema es el alto costo de las herramientas de software requeridas, pero por otro lado reduce y simplifica el diseño y la implementación en hardware. Por otra parte su utilización es de las opciones más baratas en el mercado.

Descripción

El sistema utilizado es el DSK de desarrollo TMS320C6711 (Development Starter Kit) basado en el procesador de señales digital DSP TMS320C6711, Floating-Point Digital Signal Processor de Texas instrument' (Fig. 5.13). Éste es un procesador digital de señales (DSP) de uso general que cuenta con capacidad de manejar imágenes y video; con tecnología RISC y 256 pines es una maquina potente que maneja diferentes periféricos y dispositivos internos, adicionalmente cuenta con un conjunto de periféricos externos como es memoria RAM de trabajo, convertidores analógico digital, digital analógico y puertos. El software de desarrollo utilizado es el 'Code Composer Studio' (CCS) proporcionado con el DSK; ésta herramienta permite utilizar lenguaje C de alto nivel y ensamblador, también incluye un constructor de BIOS que permite configurar los servicios de interrupciones y periféricos que se necesiten en un determinado programa.



Fig. 5.13 Procesador de señales digital.

Implementación

Trasladar las funciones y algoritmos descritos en la sección 6.2 resulta relativamente sencillo. Solo se tienen que escribir en lenguaje C cuidando las siguientes observaciones:

- Debido a que se tiene que transferir el programa al sistema se tiene que incluir las bibliotecas de funciones que permiten el traslado de los datos desde la computadora y hacia ella.

- Aunque el sistema puede trabajar con las bibliotecas estándares de C estas son muy lentas para transmitir y recibir la información como un programa común de computadora, por lo que es necesario implementar los servicios de interrupción y periféricos propios para establecer un enlace de intercambio de datos entre el sistema y la computadora.
- El espacio de memoria utilizado cambia con cada programa pero además se debe de incluir el espacio necesario para almacenar la imagen, el resultado y los parámetros necesarios para aplicar el algoritmo. Estos parámetros se indica en la compilación y enlace para generar el programa.
- Se debe de compilar y cargar cada programa nuevo o modificado antes de realizar la operación de preprocesamiento.
- Se comprobó que es más practico programar los algoritmos para trabajar con un flujo de información que con la carga total de una imagen.

La forma en que operan los algoritmos en el sistema basado en el DSP es mostrada en la figura 5.14 en la cual se observa la implementación del preprocesamiento sobre las imágenes. Se puede consultar los programas de cada algoritmo implementado en [[Anexo\2\) Sistema de preprocesamiento\Algoritmos](#)].

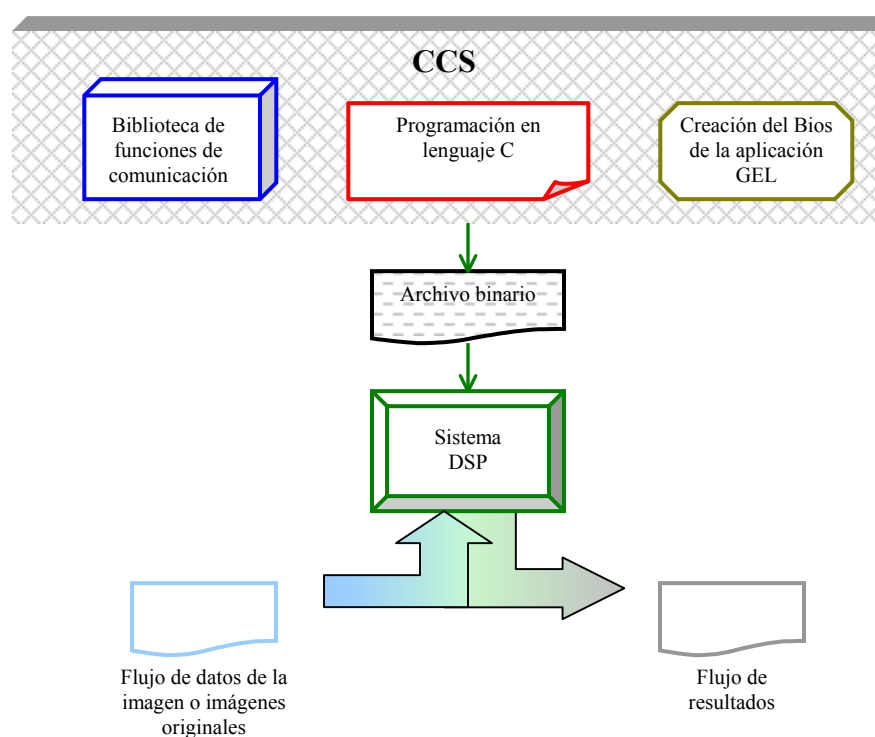


Fig. 5.14 Operación del módulo de preprocesamiento con el sistema basado en el procesador de señales digital o DSP.

Para una completa referencia de la programación e implementación y archivos para futuros trabajos con el CCS se puede consultar la bibliografía descrita en [URL5] y [[Anexo\2\) Sistema de preprocesamiento\Texas Instruments](#)].

CAPÍTULO 6: VISUALIZACIÓN DE RESULTADOS

6

Para la etapa de visualización se tuvo la necesidad de mostrar las imágenes y los resultados de forma conveniente, como opción más factible se hizo uso de una computadora (como opciones a futuro se puede utilizar dispositivos para volver a convertir los datos a una señal de video, presentarlos en una pantalla de cristal liquido, etc.). Al usar la PC se tiene una mejor forma de administrar y disponer de las imágenes. Para esto se requirió construir un programa de software que realizó las tareas de leer los datos originales y resultantes, servir de enlace ente el circuito de captura y el sistema de preprocesamiento mediante el manejo de los datos y presente en monitor las imágenes correspondientes para su apreciación.

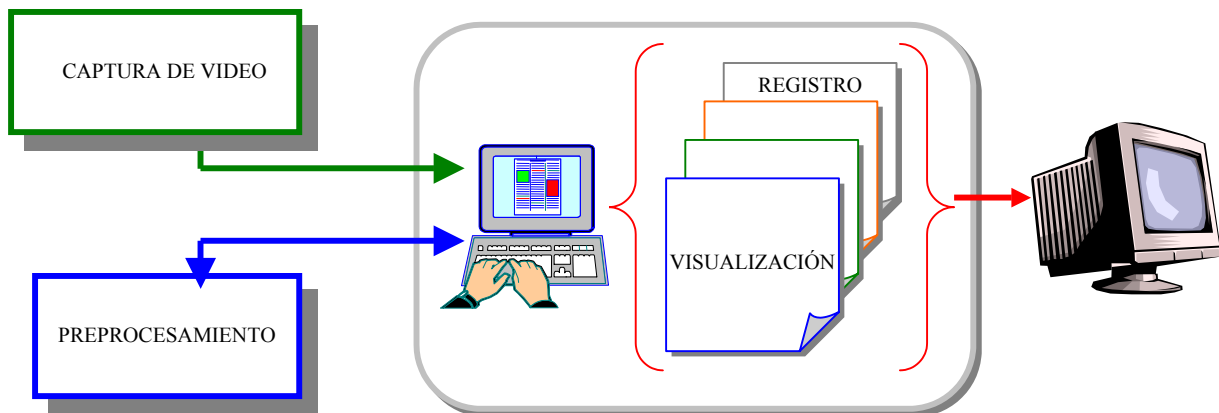


Fig. 6.1 Diagrama de flujo del programa de visualización de resultados.

Para la realización del programa se tiene una variedad de herramientas de software, de las cuales se debe seleccionar aquella que tenga las mejores capacidades de construir las funciones requeridas por el sistema en una determinada plataforma (para el caso de la tesis se buscó que el programa pudiera funcionar sobre el sistema operativo Windows de una PC). Otro factor importante en la selección de la herramienta de software fue el conocimiento del lenguaje de programación y la experiencia en su manejo. La opción utilizada en este caso fueron el lenguaje ensamblador y el lenguaje C con ambiente visual de programación.

6.1 Requerimientos del programa de visualización

De acuerdo a los requerimientos del circuito de captura y el sistema de preprocesamiento y debido a que no todo puede ser totalmente independiente al llevarlo a la práctica, se determinó que el

programa construido incluyera la mayor cantidad de particularidades del sistema total; esto se hizo así porque el programa de software es más fácil de modificar o en dado caso de ser sustituido por otro que realice las funciones que se requieran y más. Por lo tanto para el caso de esta tesis se asignaron las siguientes funciones:

- Inicialización: Programación del decodificador de video, inicialización de punteros de la memoria y funciones de transferencia de datos.
- Lee imagen: Hace peticiones al circuito de captura y extraer la imagen de la memoria.
- Manejo de datos: Asignar tipo y formato, administrar todos los datos que conforman una imagen, así como su almacenamiento.
- Visualización: Representar en forma gráfica de los datos (la imagen).
- Compatibilidad: Capacidad de exportar la imagen a un tipo de dato reconocido por otros programas de software.

La herramienta de software seleccionada para la construcción del programa de visualización fue la de 'C++ Builder' de Borland, ya que cumple ampliamente con las características requeridas como son: Ambiente visual fácil y flexible, programación conjunta de alto y bajo nivel (C++ y ensamblador), Orientación a la plataforma del sistema operativo Windows y su disponibilidad en el centro de investigación en computación, donde se realizo la tesis.

6.2 El programa

Con el paradigma de programación orientado a objetos se definieron 3 objetos principales que conforman el programa: a) Objeto 'I2C', encargado de leer y programar el decodificador de video; b) Objeto 'FIFO', éste controla la petición y lectura del contenido de la memoria para la extracción de la imagen; c) Objeto 'PROIMG', programa principal que administra las imágenes tanto del sistema de captura y el preprocesamiento. Dentro del propio objeto 'PROIMG' se incluyen otros objetos que sirven para la administración de los archivos (las imágenes) y la interfaz de usuario.

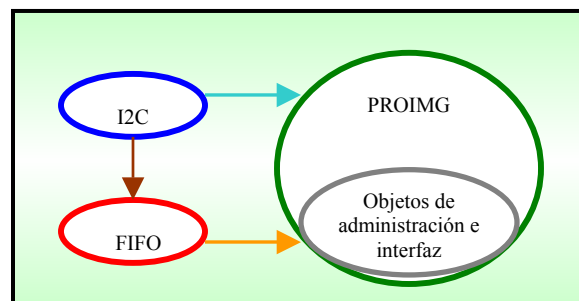


Fig. 6.2 Modelo de objetos del programa de visualización.

A continuación se muestra un esquema de la operación del programa y de las diferentes funciones que ofrece al usuario para poder realizar desde la computadora: la captura de imágenes, el preprocesamiento, su visualización y la de los resultados, así como las funciones para depositarlas en

archivos y administrarlas dentro de la PC. El programa también permite exportar los archivos de las imágenes a archivos de tipo ‘BMP’ para que puedan ser reconocidas y utilizadas por otros programas de software como el ‘Paint’ o ‘Corel-photopaint’ entre otros.

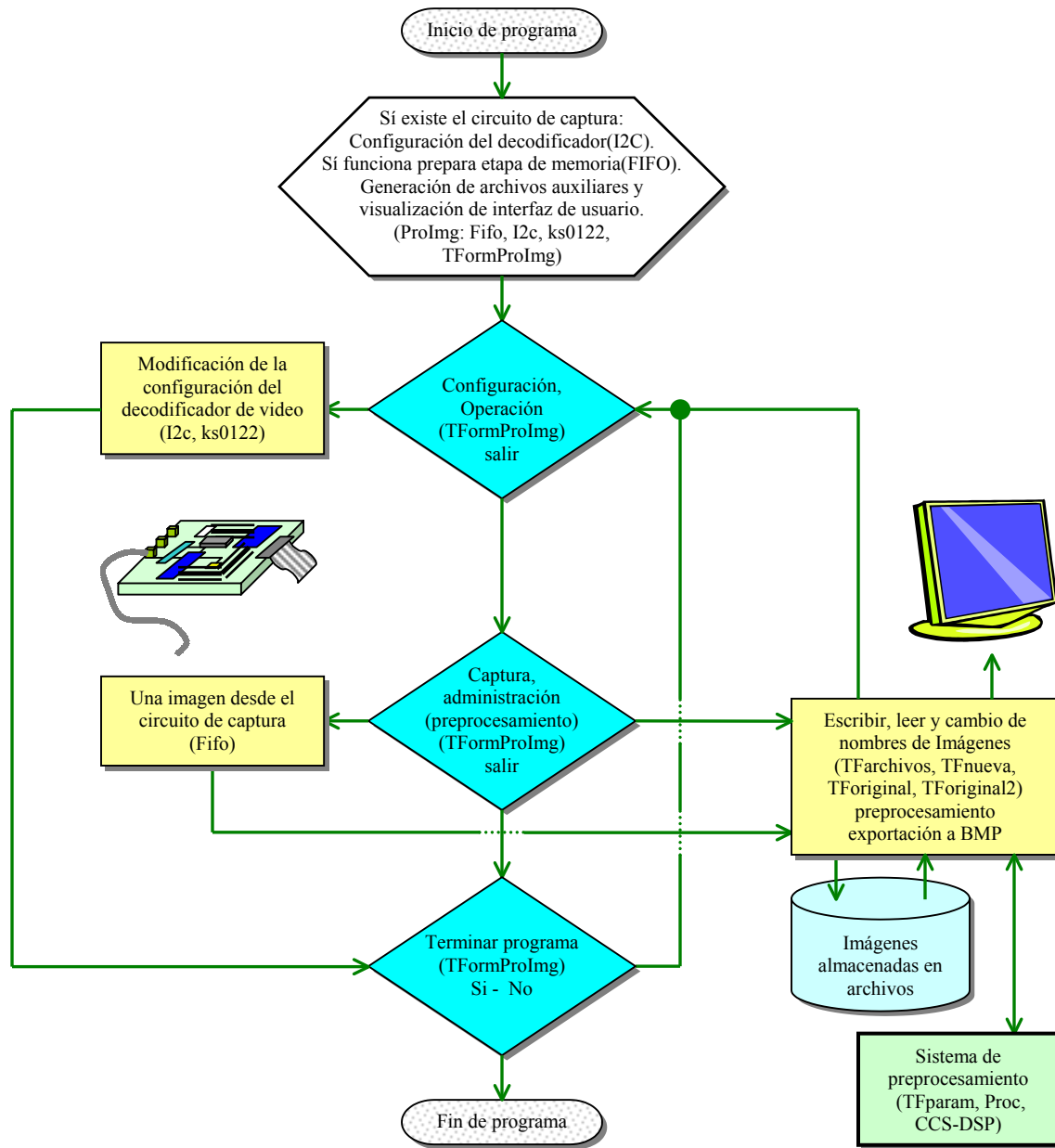


Fig. 6.3 Esquema de operación del programa de visualización.

El programa se diseño de tal manera de que al iniciar se verifica la existencia del circuito de captura y se comprueba su correcto funcionamiento, si éste no se encuentra presente o no responde a los comandos, se dan mensajes de advertencia y se procede a bloquear todas las áreas correspondientes

a la captura y a la comunicación, dejando solo las demás opciones como es la administración de imágenes previamente almacenadas y la aplicación de algoritmos de preprocesamiento. El usuario puede controlar todos los parámetros del circuito de captura, esto se realiza mediante la configuración del decodificador de video con el fin de modificar diferentes opciones de captura y así obtener mejores imágenes. Además de realizar la captura de una imagen y presentar en pantalla, esta operación puede realizarse cuantas veces sea requerido para ajustar la cámara, la iluminación o el ángulo de enfoque. Por otra parte como opciones en pantalla se tiene la administración de las imágenes capturadas en las que se puede almacenar en forma de archivos en un determinado directorio y con cualquier nombre ('____'.img) que el usuario desee, también puede sobrescribirse en un archivo ya existente y ser exportado a un archivo de tipo BMP. La salida a pantalla se realiza por medio de ventanas independientes que pueden mostrar en cada una de ellas hasta tres imágenes; una para la imagen recién capturada o como primera opción para leerse desde archivo, otra exclusiva para leer desde archivo y como segunda opción y por último una para el resultado de la aplicación de los algoritmos de preprocesamiento por medio de software (Las imágenes generadas por el sistema de preprocesamiento independiente son colocadas como un nuevo archivo). Finalmente al cerrar el programa verifica si se han guardado las posibles imágenes capturadas o preprocesadas y termina los objetos que conforman el programa (aplica los destructores necesarios para liberar la memoria y demás recursos utilizados por la PC).

6.2.1 Estructura de archivos del programa

El programa de visualización se compone de: PROIMG, un proyecto 'ProImg.bpr', que contiene las funciones principales en los archivos 'proImg', 'monitor', 'ImgOrg', 'ImgNew', 'ImgOrg2', 'archivos' y 'preproc' con extensiones 'dfm', 'cpp' y 'h' manejan toda la parte visual, como la que se ve en la presentación y características del programa. Se tiene un objeto denominado I2C contenido en los archivos 'protoi2c' con extensiones 'cpp', y 'h' que controlan todo el protocolo de comunicación entre la computadora y el decodificador de video. El objeto FIFO compuesto por los archivos 'fifo' extensiones 'cpp' y 'h' se encargan de realizar la transferencia de la imagen de la memoria del sistema de captura a un archivo que es creado en la computadora. Como parte del sistema de preprocesamiento con la computadora el objeto PROC compuesto por los archivos 'proc' extensiones 'cpp' y 'h' conforman el conjunto de algoritmos utilizados en la prueba del diseño propuesto en la tesis.

Los demás archivos se componen de la siguiente manera: un objeto hijo que hereda de I2C y FIFO para conformarse como el manejador del circuito de captura y se compone de los archivos 'seks0122' extensiones 'cpp' y 'h'. Los archivos de imágenes se identifican con la extensión 'img' que es para este caso en particular se define como un tipo de dato exclusivo del programa final. Un archivo 'bmp.ime' que contiene un formato para convertir los archivos de imágenes capturados a imágenes de tipo mapa de bits. Los archivos adicionales se generan por el proceso de compilación dando como resultado final el archivo ejecutable 'ProImg.exe' para realizar la tarea de la visualización de resultados (Todos estos archivos se encuentran en el anexo [Anexo3] Programa de visualización\Programa PROIMG]).

6.2.2 Presentación y características del programa

El programa se compone de una ventana de control, tres para la visualización de las imágenes y dos pequeñas tipo dialogo para la administración y la aplicación de los algoritmos de preprocesamiento vía software. A continuación se muestran los fólderes que conforman la ventana del programa principal (El texto de programación se da en el Anexo: [Anexo\3) Programa de visualización\Programa PROIMG\monitor.cpp y ProImg.cpp]).

La ventana principal se compone de 6 fólderes que están divididos por las funciones que desempeñan. Los 4 primeros están dedicados a la configuración del circuito de captura, más específicamente al decodificador de video. El quinto fólder nos permite capturar una imagen desde el circuito de captura y nos ofrece la posibilidad de guardara en un archivo. Finalmente se anexa un fólder extra que contiene una colección de los algoritmos de preprocesamiento generados por software.

El primer fólder del programa permite seleccionar una de 6 entrada de video que están disponibles, seleccionar la calidad de la señal de video que se utilizara como fuente y el algoritmo general que se utiliza para la decodificación de la señal de video. Cuenta con un botón de reinicio en caso de que se requiera repetir el proceso de configuración inicial que se hace al iniciar el programa. También se como datos informativos el formato de la señal, la frecuencia, el sistema de transmisión y si es a colores.



Fig. 6.4 Fólder de inicialización del programa de visualización.

El segundo fólder esta dedicado a los parámetros de salida, nos muestra el formato de salida el cual puede modificarse para obtener diferentes tipos de imágenes (RGB, YUV), Los parámetro más importantes que pueden modificarse son el brillo, el contraste y para el caso en color además se tiene

la saturación y el tinte. Otros parámetros están relacionados con la separación de luminancia y crominancia, así como el tipo de muestreo de los píxeles.

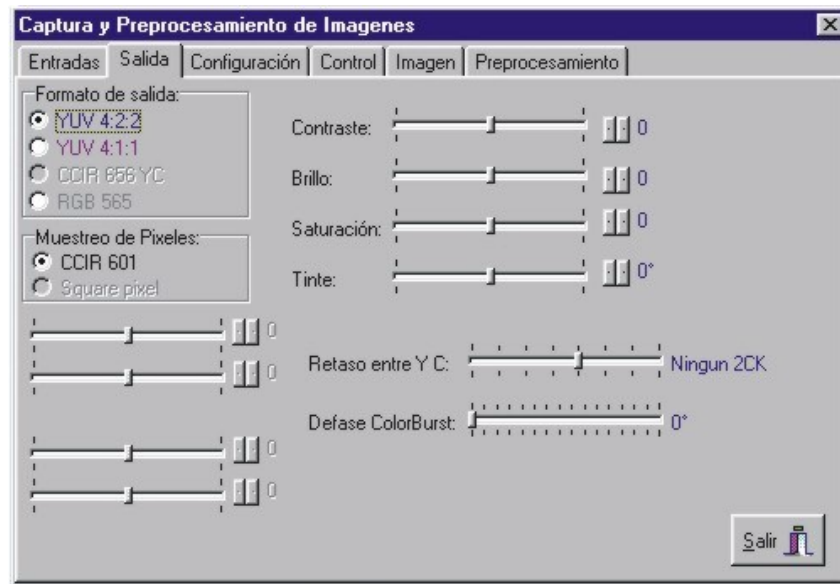


Fig. 6.5 Fólder de salida, encargada del formato de salida de la imagen.

El fólder tres es el de configuración y permite modifica diversos parámetros relacionados con los circuitos internos del decodificador de video que determinan la forma en que se extrae la información, los filtros de separación de luminancia y cominancia, así comp. el tipo y las frecuencias de corte.

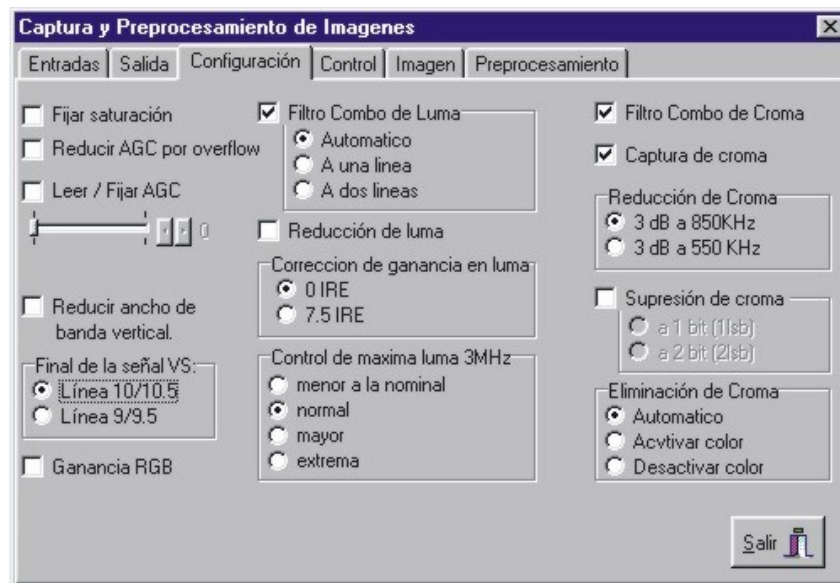


Fig. 6.6 Fólder de configuración, modifica diversos parámetros de la decodificación.

El fólder de control es el cuarto y último de la configuración y tiene que ver con el modo de operación del decodificador y de la alimentación de energía. Desde esta parte se puede inhibir el circuito, aislarlo, suspenderlo (apagarlo internamente) o cambiar la frecuencia de trabajo.

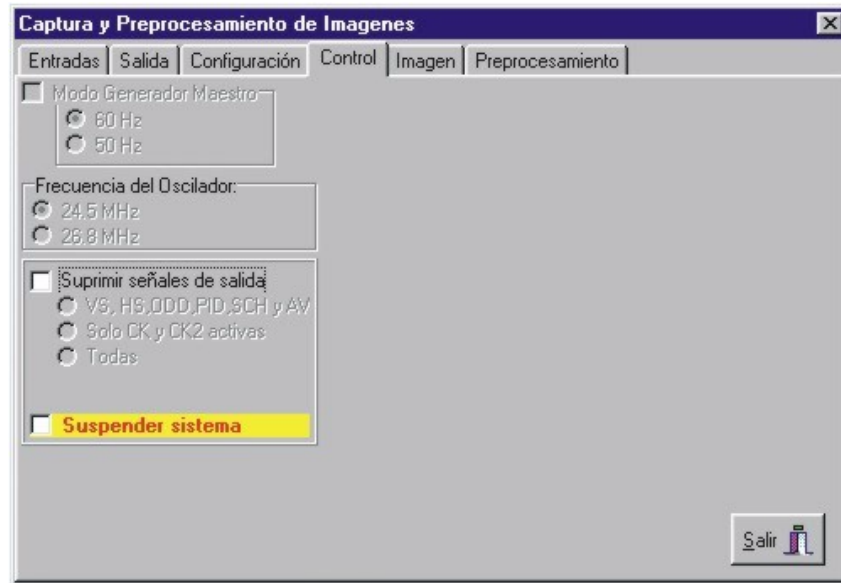


Fig. 6.7 Control del programa, modifica el modo de operación del decodificador.

El fólder de imagen esta compuesto de tres botones, el principal realizar el proceso de captura y es identificado por la imagen de una cámara, para el guardado se tiene uno con una flecha hacia un archivo y para abrir las imágenes desde archivo se utiliza el que tiene la flecha hacia fuera.



Fig. 6.8 Fólder de captura de imágenes.

Finalmente se anexo un fólder extra que permite la ejecución de los algoritmos de preprocesamiento por software. Como se ve en la figura 6.9, solo se presentan unos cuantos algoritmos, pero como una puerta a trabajos futuros, a esté se le pueden agregar más y compilar un programa con más opciones.



Fig. 6.9 Fólder de algoritmos de preprocesamiento.

Además de las pantallas principales se tienen otras que ayudan a estas con la administración de las imágenes y las que presentan las propias imágenes ya sea capturadas o leídas desde un archivo.

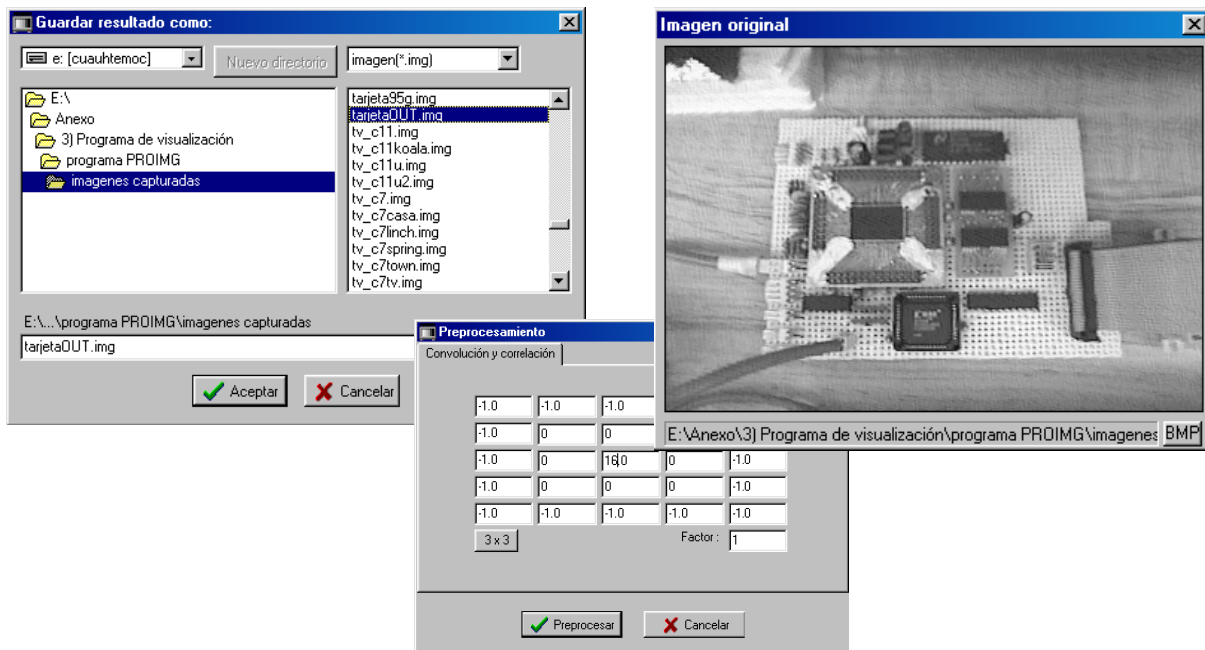


Fig. 6.10 Ventanas de administración, visualización y de parámetros de preprocesamiento.

CAPÍTULO 7: RESULTADOS Y CONCLUSIONES

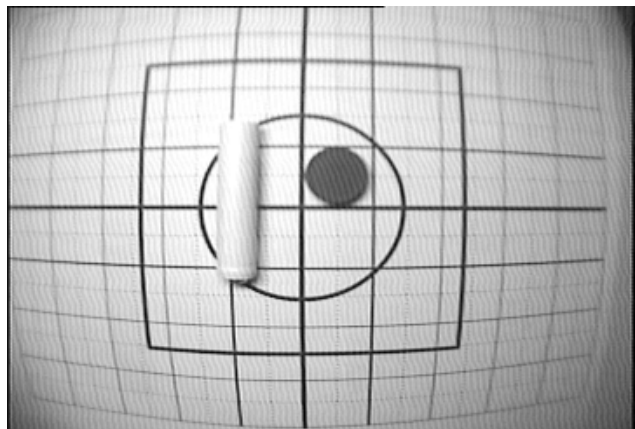
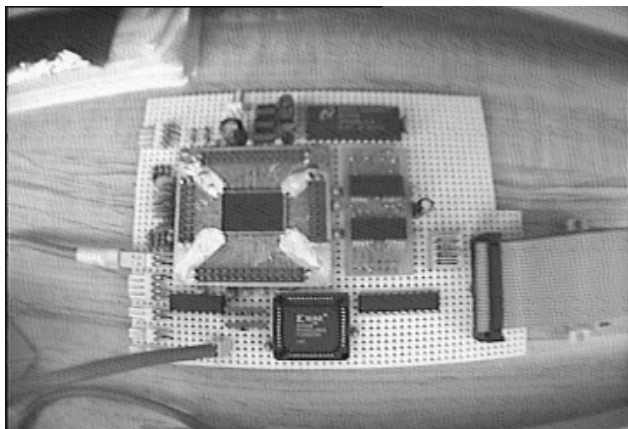
7

7.1 Obtención de resultados

7.1.1 Captura de imágenes

Se construyó un prototipo basado en el circuito diseñado y descrito en la sección 4.2 del capítulo ‘Captura de imágenes’ para realizar la captura. Se creó un sistema de captura que es capaz de leer video en tiempo real proveniente de cualquier fuente de video ya sea compuesto, s-video o RGB. A la salida entrega un flujo de imágenes en formato ‘Y R-Y B-Y’ donde tomamos sólo la parte de luminancia (Y) para obtener imágenes en escala de grises con resolución de 8 bits. El flujo de las imágenes está disponible en la memoria temporal que se actualiza 30 veces por segundo. La obtención de una imagen se realiza deteniendo por un instante el flujo, se lee y transfieren los datos presentes al sistema anfitrión (la computadora personal) y se vuelve a continuar capturando. Los datos de las imágenes tomadas del circuito de captura se pueden almacenar en forma de archivos por el anfitrión para su posterior manejo y manipulación. Un diseño de circuito impreso mejorado está disponible en el Anexo [📎 Anexo\3) Programa de visualización\Programa PROIMG\imagenes capturadas\tarjeta.img].

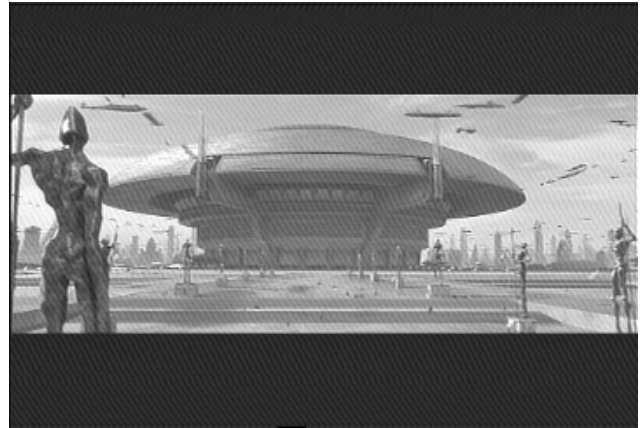
Como resultado se obtienen imágenes digitales en escala de grises. Para explicarlas es mejor mostrarlas para así ver los resultados. A continuación se muestra una serie de imágenes provenientes de diferentes dispositivos y aparatos que generan señales de video (TV, Cámara de video, VCR, DVD, etc.); para más ejemplos puede revisar los archivos con extensión ‘img’ disponibles en el Anexo [📎 Anexo\3) Programa de visualización\..\imagenes capturadas] (Nota: Utilice el programa de visualización).



Figs. 7.1 y 7.2 Imágenes provenientes de una mini cámara de vigilancia que muestra al propio prototipo realizando la captura y a un par de objetos sobre un patrón geométrico.



Fig. 7.3 Imagen de televisión publica y Fig. 7.4 Imagen de televisión restringida.



Figs 7.5 y 7.6 Imágenes de películas generadas por un reproductor de DVD.

7.1.2 Preprocesamiento de imágenes

Las funciones y algoritmos tienen como fin obtener características, información u otros datos necesarios para un determinado propósito, ejemplo de esto es destacar el contorno de un objeto o la eliminar información no necesaria. Los resultados generados por las funciones y algoritmos definidos en la sección 5.2 del capítulo de 'Preprocesamiento de imágenes' son mostrados a continuación dando la imagen original y sus resultados.

Umbralizado.- Al aplicar la función en la imagen se identifican las áreas correspondientes a la parte del cielo y la de los edificios divididas por el horizonte.



Fig. 7.7 Imagen capturada original 1.



Fig. 7.8 Imagen binarizada con un umbral de niveles de gris igual de 137.

Operaciones básicas.- Podemos separar, adicionar y combinar información de dos imágenes. A continuación se muestran los resultados al aplicar las funciones sobre estas:

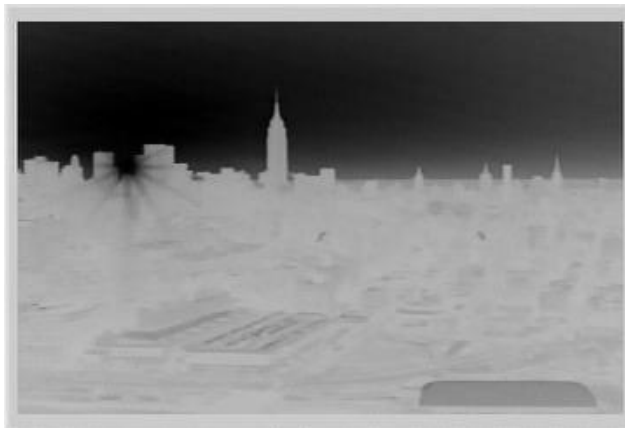


Fig. 7.9 Función NOT sobre la figura 7.7.



Fig. 7.10 Función Máximo entre las figuras 7.7 y 7.8.



Fig. 7.11 Función Mínimo entre las figuras 7.7 y 7.8.



Fig. 7.12 Función Resta entre las figuras 7.7 y 7.8.



Fig. 7.13 Función Suma Absoluta entre las figuras 7.7 y 7.8.



Fig. 7.14 Función Suma Promedio entre las figuras 7.7 y 7.8.



Fig. 7.15 Función AND entre las figuras 7.7 y 7.8.



Fig. 7.16 Función OR entre las figuras 7.7 y 7.8.



Fig. 7.17 Función XOR entre las figuras 7.7 y 7.8.

Transformación geométrica.- Algunas fuentes de video entregan imágenes que tienen una distorsión geométrica, ya sea por el ángulo en que se enfocó o por efecto de la propia lente, en este caso se tiene una imagen que muestra la vista perpendicular de una superficie esférica debido a la lente por lo que se tiene que realizar una transformación que extienda esa superficie en un plano; usando el algoritmo descrito en el capítulo 5 tenemos:

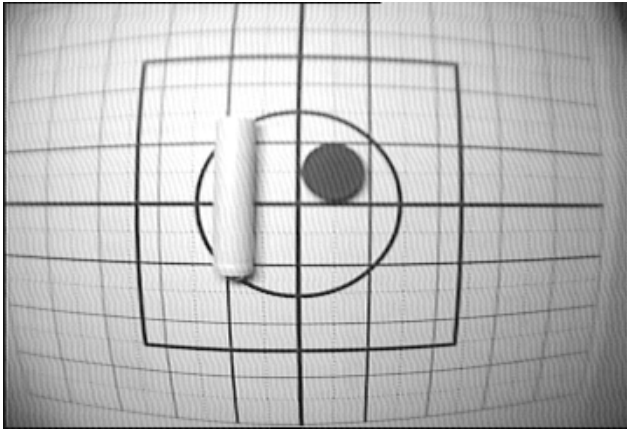


Fig. 7.18 Imagen capturada original 2.

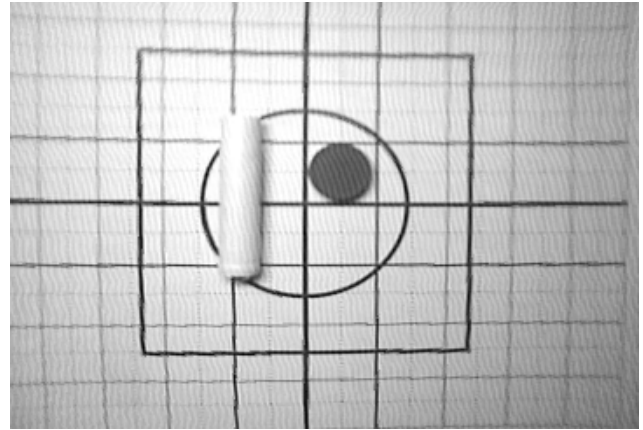


Fig. 7.19 Transformación de una superficie esférica a una plana.

La convolución espacial es muy útil pero con este tipo de algoritmo también se aplica para realizar diferentes tipos de transformaciones como es un filtro pasa-bajos etc. Los siguientes ejemplos son obtenidos a partir de las ventanas o mascarar siguientes:

Filtro espacial



Fig. 7.20 Imagen capturada original 3.

Pasa bajos

$1/25 *$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Ventana 7.1
Modelo de un filtro pasa bajos



Fig. 7.21 Convolución espacial entre la figura 7.20 y la ventana 7.1.

Pasa altos (Detector de bordes)

-1	-1	-1	-1	-1
-1	0	0	0	-1
-1	0	16	0	-1
-1	0	0	0	-1
-1	-1	-1	-1	-1

Ventana 7.2
Modelo de un filtro pasa altos.



Fig. 7.22 Convolución espacial entre la figura 7.20 y la ventana 7.2.

7.1.3 Visualización

Como resultado del programa de visualización se cuenta con las imágenes obtenidas en este documento, además se proporcionó una estructura para la administración de las imágenes, en la cual se pueden guardar, borrar, describir y buscar dentro de los dispositivos de almacenamiento del anfitrión (la PC) dentro del ambiente de Windows.

Desde un principio se pretendió realizar una interfaz de usuario lo más sencilla posible, de tal manera que la operación principal quedo reducida a la utilización de tres botones, uno para la captura, otro para aplicar el algoritmo de preprocesamiento y uno más para almacenar la imagen en forma de archivo. Se agregaron otros botones para dar mayor flexibilidad al programa como es el botón de reinicio, el de abrir una segunda imagen y el de exportar a BMP. Con poca práctica cualquier persona puede conocer y utilizar todos los componentes que se encuentran en el programa y enfocarse en la tarea del preprocesamiento de imágenes. El programa fue realizado para una plataforma Windows 98 con lenguaje C++ pero se ofrecen todos los archivos fuente que pueden ser utilizados para generar una compilación en plataformas diferentes como Windows 2000, Windows XP o Linux entre otras. La única limitante es la interfaz a la computadora que tiene que ser modificada de acuerdo a la plataforma que se quiera utilizar; una alternativa es la utilización de una interfaz estándar que sea compatible con cualquier plataforma como es la de bus serial.

Otra de las ventajas de tener los archivos de programación es de ofrecer a cualquier persona interesada la posibilidad de generar un nuevo programa, modificando, eliminando o agregando componentes y funciones de acuerdo a su propia conveniencia. También puede utilizar las bibliotecas construidas por separado dada la planificación por objetos trasladándolas a otros programas de forma sencilla.

7.2 Conclusiones, metas y logros alcanzados

Dados los resultados obtenidos en la sección anterior se puede concluir que se cumplieron los objetivos tanto general como específicos propuestos al inicio de la tesis y en el planteamiento del problema. Como resultado, se aporta el diseño de un sistema que, utilizando componentes novedosos permite realizar tareas de captura y preprocesamiento a imágenes que luego pueden ser utilizadas en

otras aplicaciones. Otro resultado buscado fue el aportar elementos básicos que permiten incursionar en nuevos proyectos donde el sistema construido sea la herramienta básica o el punto de partida, ya que al haber hecho el sistema modular puede ser rediseñado todo o cada modulo, puede ser complementado con nuevos componentes y nuevas funciones entre otras. Pero como logro tangible, el sistema es utilizado para realizar tareas primarias de preprocesamiento de imágenes.

De este punto se puede partir para realizar diferentes proyectos y aplicaciones específicas. En principio se puede utilizar en el tratamiento de imágenes para eliminar ruido, ecualizar los niveles de gris, separar e identificar regiones dentro de una determinada porción de la imagen. En visión por computadora se puede utilizar para filtrar imágenes que encuentren determinados patrones que sirvan de guía para el análisis y cálculo de una determinada trayectoria. Al segmentar y cuantificar se puede utilizar en el reconocimiento patrones.

Por otro lado, para obtener mejores resultados se concluye que los factores que se presentan durante la generación de la imagen original son de gran importancia, ya que para obtener una imagen lo más clara posible o para que resalten los detalles o características buscadas es necesario controlar el ambiente donde se origina la imagen; entre estos factores se encuentra la correcta posición de la cámara, color y textura del fondo, la intensidad, cantidad y color de la luz, esto ayuda para que la imagen resultante tenga las condiciones adecuadas que impliquen la aplicación de un número reducido de funciones y algoritmos de preprocesamiento, todo esto es de importancia cuando el tiempo también es un factor importante. En cuanto al costo y disponibilidades se tiene que el sistema es caro como prototipo, pero al producirse en grandes volúmenes los costos se reducen considerablemente.

7.2.1 Circuito de captura

La construcción y puesta a punto del circuito tomó una parte importante del tiempo ya que a pesar de ser un circuito pequeño tiene su nivel de complicación con respecto a los factores de sincronización, valores de los componentes necesarios, parámetros eléctricos, dispositivos periféricos requeridos y programación. El diseño y su implementación dieron paso a una serie de pruebas y correcciones para finalmente obtener un circuito con un desempeño optimo aceptable entendiendo que no fue nada trivial el análisis y trabajo realizado. Aunque el sistema trabaja en escala de grises tiene todas las posibilidades de capturar en color con una resolución de 16 bits.

7.2.2 Preprocesamiento de imágenes

Las funciones y algoritmos diseñados funcionaron adecuadamente, aunque la colección de estos es pequeña y esta abierta a mejoras presenta una muestra general del desempeño del sistema en su conjunto. Más que implementar todas las funciones posibles que se pudieran realizar, se ofrece un método a utilizar para que se realicen los propios siguiendo el mismo camino. De esta forma se puede ir directamente al diseño de aplicaciones específicas, diseñarlas, estructurarlas y mejorar su desempeño con este sistema. La razón de realizar el preprocesamiento en escala de grises es por que a parte de que a colores se multiplica el trabajo, la mayor parte de las funciones y algoritmos solo requieren un tipo de espectro ya sea en grises o en un determinado color específico. Con respecto al hardware a utilizar para la realización de esta etapa, se vio que es conveniente seguir los siguientes pasos: Implementar los algoritmos en software, depurarlos, probarlos y mejorarlos en un sistema cliente – servidor como lo realizado en este caso. Finalmente una vez realizado todo lo anterior y teniendo una aplicación específica se puede pasar a construir un sistema totalmente en hardware.

7.2.3 Visualización

El programa de visualización es la parte más flexible y la que costo menos realizar, debido a la experiencia en programación y sólo se le dio aspecto, presentación y eficiencia, dejando a parte las demás interfaces de usuario. Sólo la etapa de comunicación con las otras partes generó una serie de análisis, diseño y pruebas para conseguirlo exitosamente, optándose por la forma más sencilla ya que implicaba un mayor tiempo para el diseño de controladores (drivers) que se encargaran de la función.

Aunque el sistema es pequeño se diseñó de forma tal que puede ser ampliado y mejorado usando toda la documentación disponible en el Anexo. Parte de todas las posibilidades que se pueden realizar son explicadas en la siguiente sección.

7.3 Mejoras y trabajos a futuro

A continuación se describen mejoras que se pueden realizar al sistema aprovechando las posibilidades y características que se incluyeron en el diseño actual.

Realizar la captura a color, sólo se tiene que duplicar la cantidad de memoria temporal; aunque es difícil de conseguir es posible sustituirlas por otras, pero se debe de considerar los tiempos de trabajo de escritura y lectura, se sugiere que sean de 15ns o menos. Cuadruplicando la cantidad de memoria pueden obtenerse imágenes del doble de tamaño y a color.

Cambiando la lógica de control y el bus de comunicación se puede trasladar el sistema de captura de imágenes a uno de captura de video, esto se puede realizar cambiando la interfaz de comunicación a USB. De esta forma, con realizar unos ajustes en el programa de visualización puede tenerse un flujo de imágenes contrario a tomar una a la vez.

La etapa de visualización se puede mejorar para que sea más agradable al usuario. Se puede agregar funciones de conversión de archivos y se puede incluir un sistema de ayuda para que explique el funcionamiento de cada control que puede ser alterado por éste. Puede modificarse la etapa de transferencia de datos utilizando los controladores (drivers) de comunicación de los puertos USB de la computadora para complementar la idea descrita en el párrafo anterior.

Agregar más funciones que puedan ser llamadas desde el programa de visualización o por el propio sistema de preprocesamiento. Se pueden crear bibliotecas de funciones de acuerdo con el tipo y utilizarse en la creación de aplicaciones específicas.

Además de un mejoramiento en el sistema se puede adecuar para tareas específicas, ejemplo de esto puede ser el utilizar el sistema de captura para la etapa de transmisión de un videoteléfono, decodificador para una pantalla planas de cristal líquido. Junto con algunos algoritmos de preprocesamiento puede servir para transformar de un tipo de señal a otro como es pasar de YCbCr a RGB y viceversa.

Estos trabajos y mejoras son resultado de ideas, posibilidades y limitantes que se fueron presentando dentro del desarrollo de la tesis.

BIBLIOGRAFÍA

B

A continuación se muestra la bibliografía escrita y las referencias electrónicas principales que se consultaron y utilizaron para la realización de la tesis. Se seleccionaron estas por su relevancia, ya que al revisar un amplio catalogo, en general la información encontrada es similar.

- [1] **Charte Ojeda Francisco., ‘Programación con C++ Builder 5’**, Aprenda a desarrollar servidores y clientes COM y COBRA. Edición 2000.

El libro proporciona los elementos necesarios para la programación en C++ y las extensiones existentes en C++ Builder. Detalle sobre la creación de componentes VCL, editores de propiedades y de componentes. Manejo de bases de datos, exportación y creación de directorios de datos propios. Etc.

Editorial: Anaya Multimedia (Grupo Anaya S.A) ISBN 8-44-151046-6.

- [2] **González Rafael C./ Woods Richard E., ‘Tratamiento digital de imágenes’**, Versión en castellano de Digital Image Processing. Tercera Edición 1996 y 1992 en Inglés.

El libro proporciona una introducción a los conceptos y metodologías básicos del tratamiento de imágenes y desarrolla las bases para unos estudios más amplios en este campo, de continua y rápida evolución. Los temas tratados abarcan desde la mejora y restauración hasta la codificación, segmentación, descripción, reconocimiento e interpretación de imágenes.

Editorial: Addison-Wesley Iberoamericana, S.A. ISBN: 0-20-150803-6.

- [3] **Keith Jack, ‘Video Demystified’**, *A Hardware book for the Digital Engineer*, Second Edition HARRIS Semiconductors. 1996, en Inglés.

La referencia estándar de video para ingenieros digitales con información de referencia a la mano cubriendo todos los estándares internacionales, así como el diseño de arquitecturas para la incorporación de video dentro de sistemas computacionales, video conferencia, técnicas de codificación y decodificación MPEG etc.

Editorial: HighText publications, ISBN: 1-87-870736-1.

- [4] **Kenneth R. Castelman, 'Digital image Processing'**, 1996, en Inglés.
Debido al explosivo crecimiento del procesamiento de imágenes sobre todo en las últimas décadas, este interés se ha dado por la curiosidad acerca de los video juegos y efectos especiales de la industria del entretenimiento. Sin embargo el impacto más significativo del procesamiento de imágenes se ha dado en aplicaciones para resolver problemas del mundo real.
Editorial: Prentice hall ISBN: 0-13-211467-4.
- [5] **Poynton Charles, 'Introduction to Digital Video'**, A summary of the fundamental concepts of digital video. 1996, en Inglés.
Referencia para conocer si es que no se tiene familiaridad con el concepto de video y su entorno o para refrescar el conocimiento para comprender el manejo del video.
Editorial: John Wiley & Sons. ISBN: 0-47-112253-X.
- [6] **Rhys Lewis, 'Practical Digital Image Processing'**, Ellis Horwood series in digital and signal processing. 1990, En Inglés.
Recopilación de textos y conferencias de especialistas que integran las operaciones y funciones matemáticas orientadas al procesamiento de imágenes.
Editorial: Ellis Horwood. ISBN: 0-13-683525-2.

REFERENCIAS BIBLIOGRÁFICAS EN INTERNET:

- [URL1] **PHILIPS, I2C Specifications.**
Especificaciones del protocolo I2C y su implementación
<http://www.philips.com/>
<http://people.freebsd.org/~nsouch/iicbus.html>
<http://www.semiconductors.philips.com/buses/i2c/>
- [URL2] **Innovative Tech Works.**
Empresa dedicada a la fabricación de tarjetas de captura de imágenes y video.
<http://tvland.maxinet.com/itworks>
- [URL3] **LM1881 video sync separator.**
Hoja de especificaciones del circuito de separación de señales de video y comunicación de National Semiconductors
<http://www.national.com/pf/LM/LM1881.html>
- [URL4] **KS0122-S, S5D0127X01 - KS0127 video decoders.**
Hoja de especificaciones del decodificador de video de Samsung Semiconductors
<http://www.usa.samsungsemi.com/products/prodspec/videodecoder/KS0122-S/ks0122ov.htm>

<http://www.usa.samsungsemi.com/products/prodspec/videodecoder/S5D0127X1/>

[URL5] **TVP5022 video decoder y DSP TMS320C6711 DSK.**

Hoja de especificaciones del codificador decodificador de video de Texas Instrument.

<http://www.ti.com/sc/docs/products/analog/typ5022.html>

Manuales del paquete de desarrollo del DSP TMS320C6711 y del Software de compilación Code Composer Studio de Texas Instrument.

http://www.ti.com/.../docs/c6000_manuals.htm

[URL6] **The Macula.**

The Macula Includes Color Vision and Amsler Grid Testing Anatomy, Physiology and Pathology of the Human Eye.

http://homes.aol.com/MonT714/tutorial/the_eye/macula.html

[URL7] **Alpha Data Parallel Systems Ltd.**

Procesamiento paralelo, multiprocesamiento, procesamiento digital de señales, procesamiento de imágenes, DSP, simulación, tiempo real, visión de maquinas, TMS320C40, C44, TMS320C44, C40 System, procesador C40, enlaces e interfaz RS232, entrada/salida digital, procesadores Alpha RISC, procesadores T9000, enrutamiento de canal virtual, VCR, etc..

<http://www.alphadata.co.uk/>

[URL8] **Camera Advisor -- Find the right camera for your vision application.**

Recomendaciones para seleccionar una cámara de video, por National Instrument.

<http://www.ni.com/imaq/>

<http://sine.ni.com/appswe/nipc.specs?action=search&asid=1102>

[URL9] **Sistemas actuales de captura, digitalización y preprocesamiento de imágenes.**

http://www.sundance.com/html/im_cap.htm

<http://www.sensoray.com/html/prodcts3.htm#framegrabber>

<http://www.dagemti.com/frmegrbr.htm>

<http://www.i-cubeinc.com/hardware-chart.htm>

[URL10] **Enciclopedia de términos de electrónica y computación en la red.**


<http://webopedia.internet.com/TERM/C>

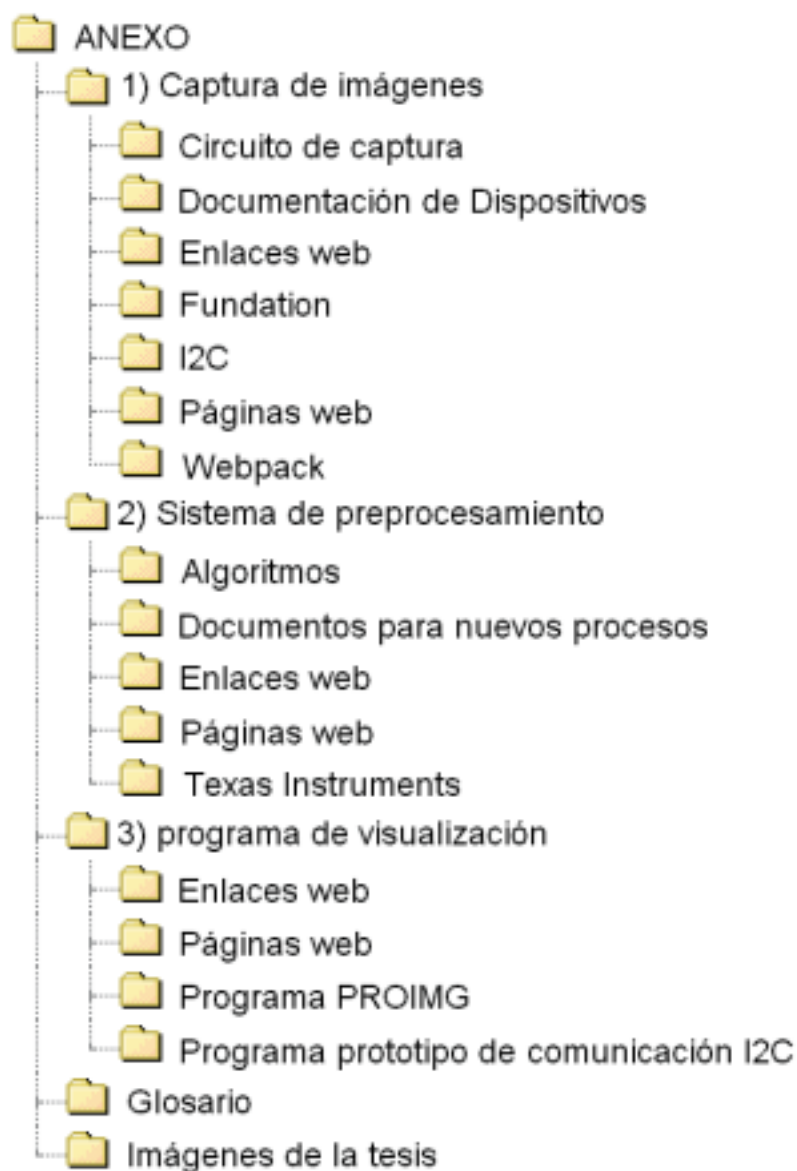
[URL11] **Memoria de video.**

<http://www.okisemi.com/>

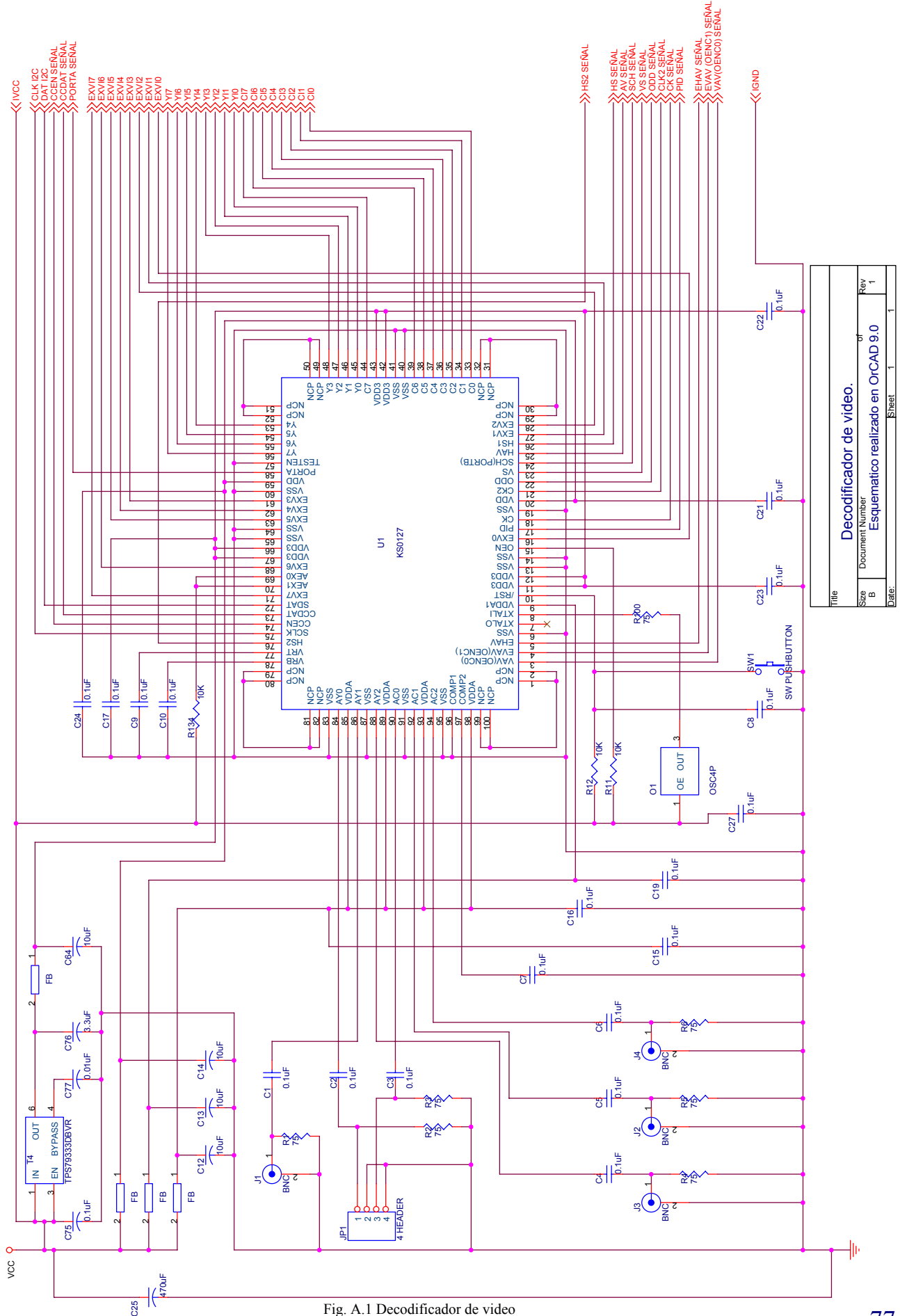
APÉNDICE

A

 Esta es una recopilación de todo el contenido y material utilizado para la realización de la tesis y se proporciona en un CD junto con este documento y se encuentra distribuido de la siguiente manera:

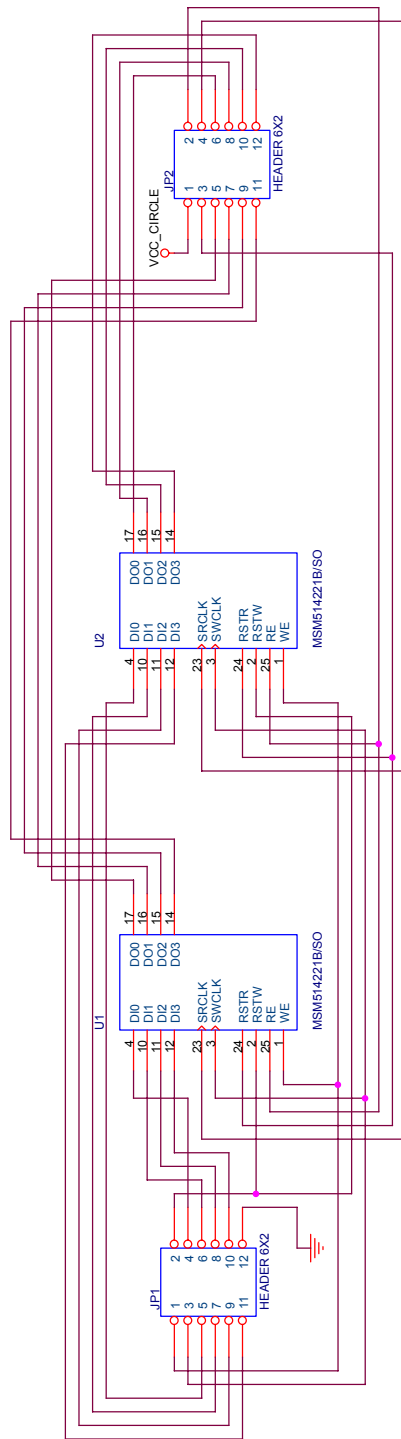


Esquemas, diagramas eléctricos, circuitos lógicos y PCB's del circuito de captura de imágenes:



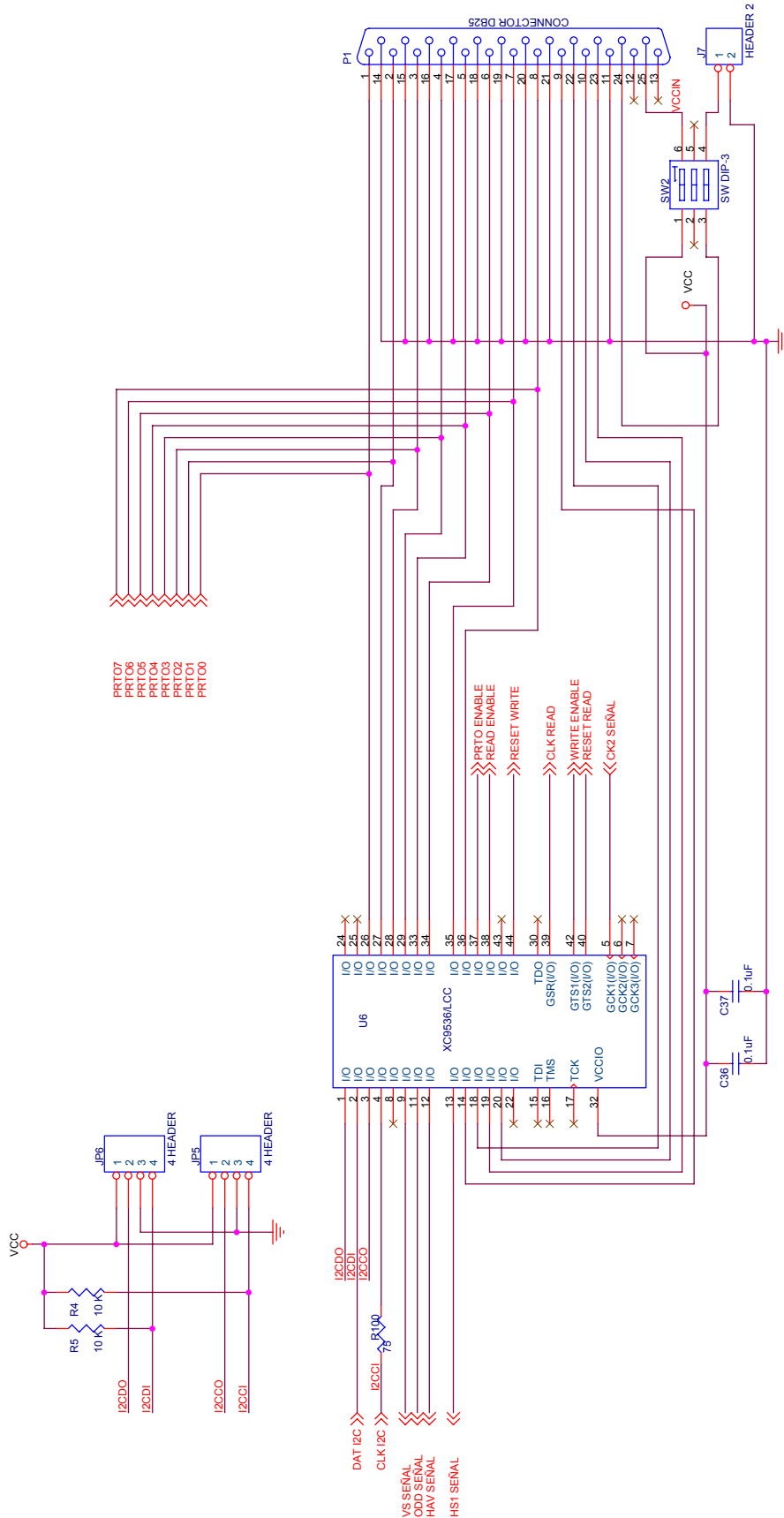
Title		Decodificador de video.	
Size	Document Number	of	Rev
B	Esquemático realizado en OrCAD 9.0	1	1
Date:	Sheet	1	1

Fig. A.1 Decodificador de video



Title		Memoria de almacenamiento de video.	
Size	Document Number	Rev	
B		1	
Date:		Sheet	1 of 1
		Esquematico realizado en OrcAD 9.0	

Fig. A 2 Memoria de almacenamiento de video



Title			
Circuito de control y comunicación de datos.			
Size	Document Number	Rev	
B	Esquemático realizado en OrCAD 9.0	1	
Date:	Sheet	1	1

Fig. A.3 Circuito de control y comunicación de datos.

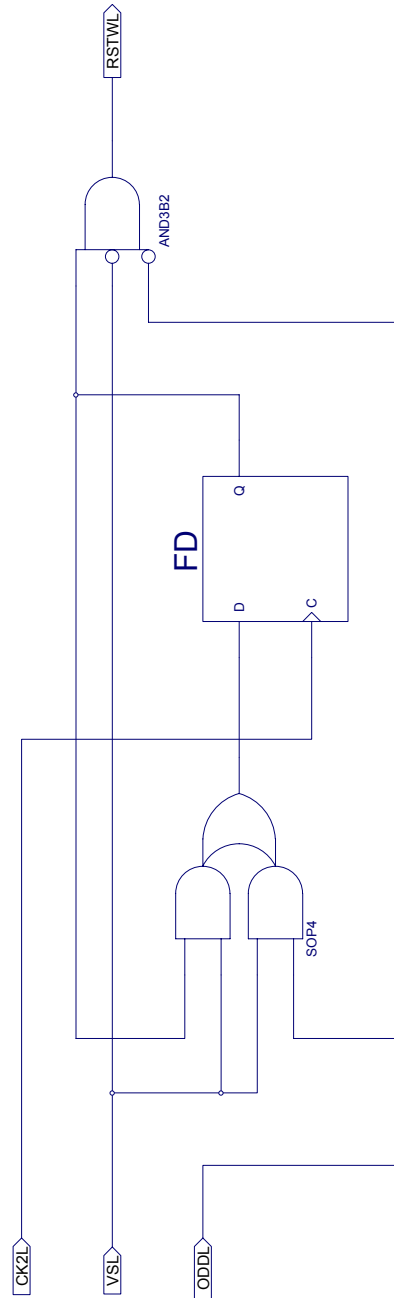


Fig. A.4 Circuito lógico de la escritura a los módulos de memoria.

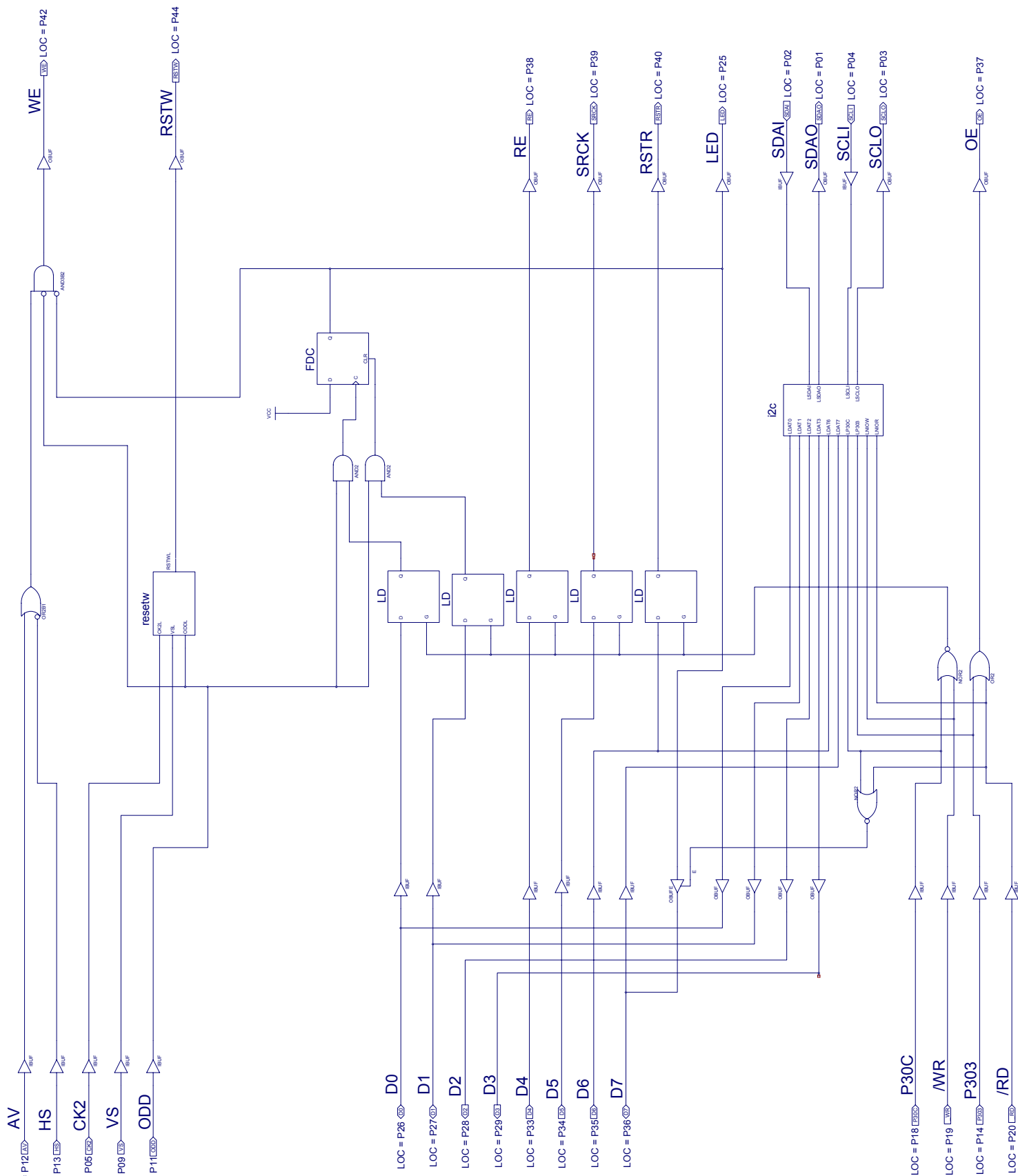


Fig. A.5 Circuito lógico de control e interfaz con el puerto de la PC.

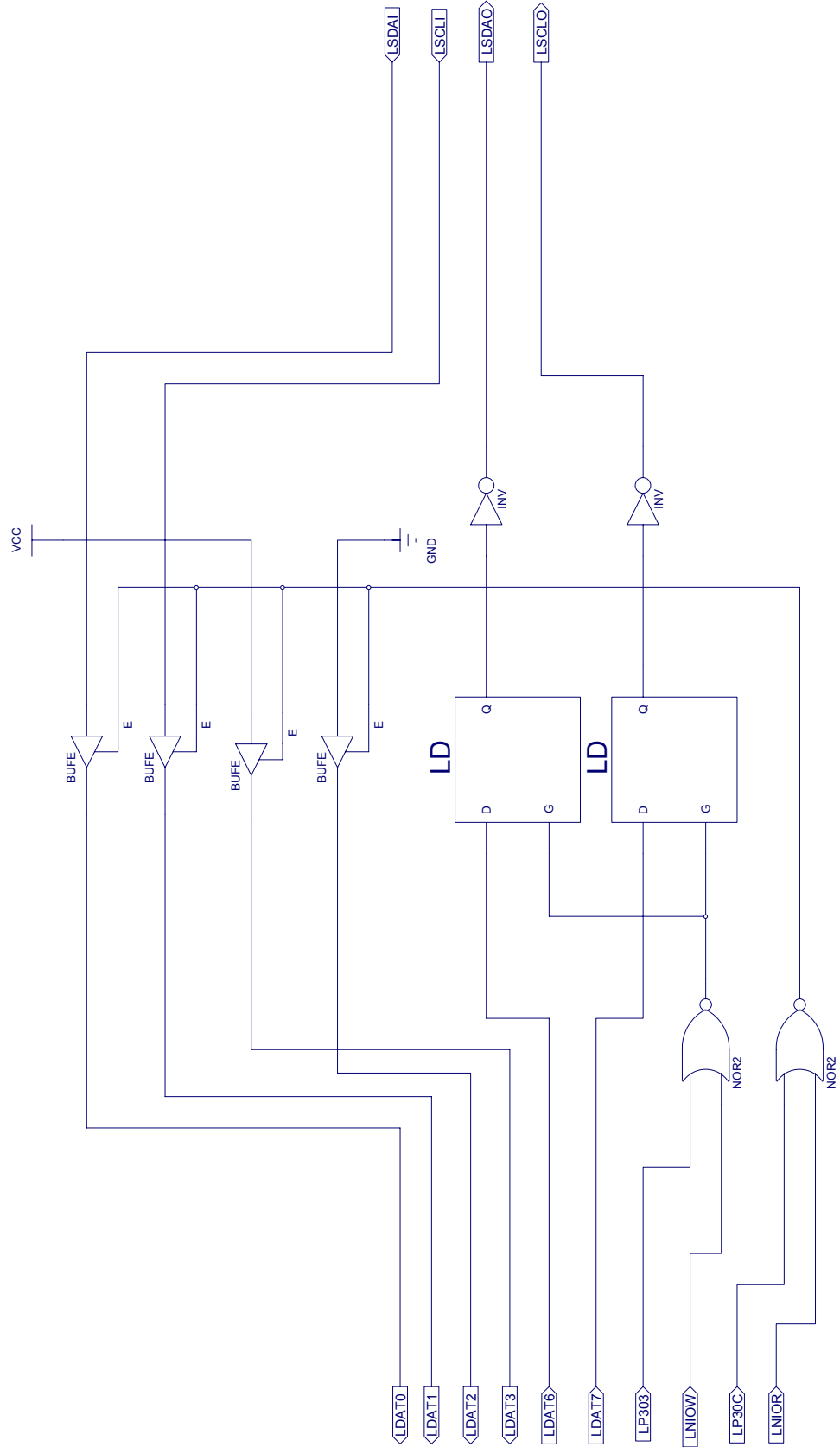


Fig. A.6 Circuito lógico del módulo I²C.

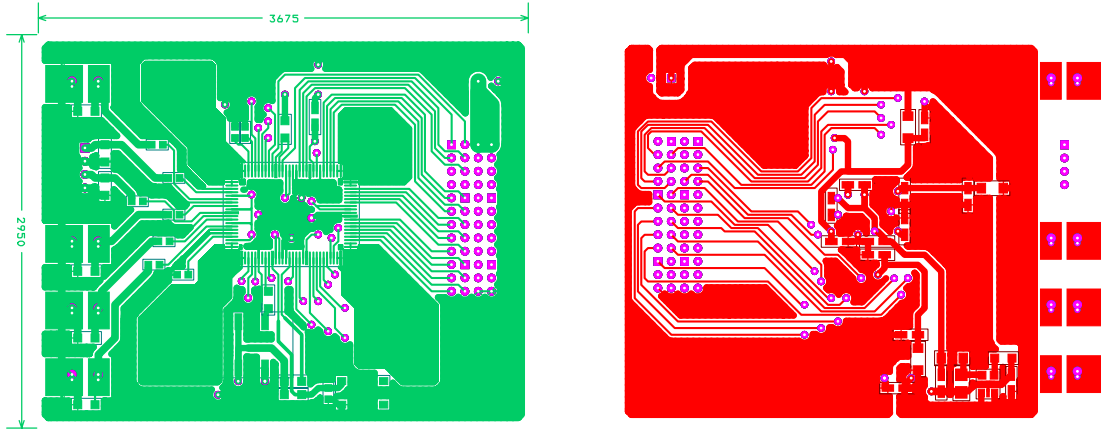


Fig. A.7 Circuito impreso 1.

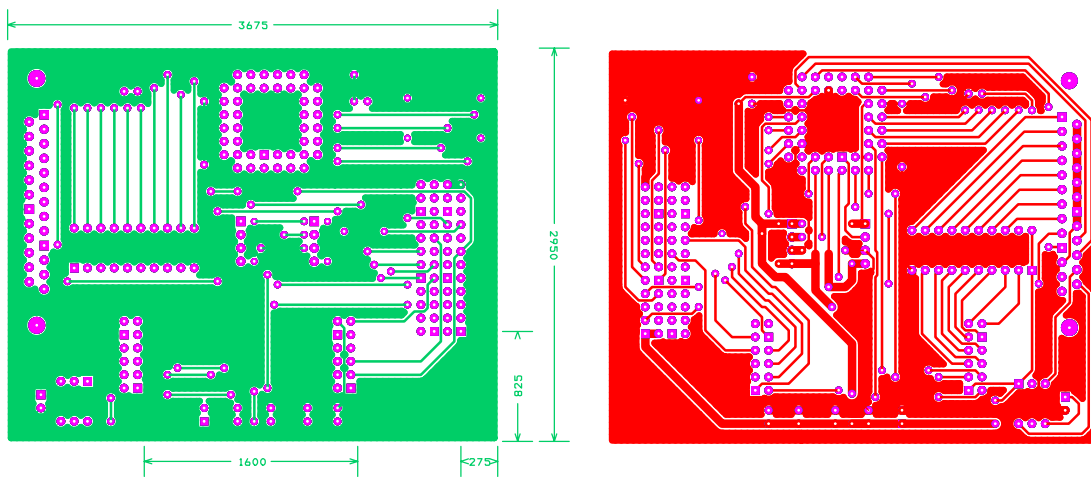


Fig. A.8 Circuito impreso 2.

Nota: Estos diagramas y circuitos también están disponibles en el CD.

