

Tecnológico de Costa Rica  
Escuela de Ingeniería Electrónica



**Diseño de ...**

Informe de Proyecto de Graduación para optar por el título de  
Ingeniero en Electrónica con el grado académico de Licenciatura

Angel Phillips Ortega

Borrador de 11 de mayo de 2014



Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Angel Phillips OrtegaCartago, 11 de mayo de 2014

Céd: 3-0443-0358



Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería Electrónica  
Proyecto de Graduación  
Tribunal Evaluador

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

---

M.Sc. Eduardo Interiano Salguero  
Profesor Lector

---

Dr. Renato Rimolo Donadio  
Profesor Lector

---

Lic. Jorge Castro Godínez  
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 29 de noviembre de 2011



Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería Electrónica  
Proyecto de Graduación  
Tribunal Evaluador  
Acta de Evaluación

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

↶ "RECORDARE"  
editar la acta

Estudiante: Angel Billy Phillips Ortega

Nombre del Proyecto: *Implementación de una aplicación para ...*

Miembros del Tribunal

---

M.Sc. Eduardo Interiano Salguero  
Profesor Lector

---

Dr. Renato Rimolo Donadio  
Profesor Lector

---

Lic. Jorge Castro Godínez  
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Nota final del Proyecto de Graduación: \_\_\_\_\_

Cartago, 29 de noviembre de 2011





# Resumen

El resumen es la síntesis de lo que aparecerá en el tesis. Tiene que ser lo suficientemente consiso y claro para que alguien que lo lea sepa qué esperar del resto de la tesis si la leyera completamente. Puede concluir con palabras clave, que son los temas principales tratados en el documento. El resumen queda fuera de la numeración del resto de secciones.

No se acostumbra utilizar referencias bibliográficas, tablas, o figuras en el resumen.

**Palabras clave:** palabras, clave, ...



# Abstract

The same as before, but in English.

**Keywords:** word 1, word 2,



*a mis queridos padres*



# Agradecimientos

Agradezco principalmente a Dios por haberme dado la sabiduría, la fortaleza y serenidad necesarias para afrontar con retos y adversidades presentes durante estos años para concluir la carrera de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

A mi madre Yolanda, por el gran cariño y apoyo incondicional que me ha permitido alcanzar grandes logros para continuar siendo una persona de bien.

Al Ing. Jorge Castro Godínez, por su apreciada ayuda y consejo durante la conclusión de mis estudios y en la realización de este trabajo.

Y a todas las personas que han contribuido de forma integral a mi formación.

Angel Phillips Ortega

Cartago, 11 de mayo de 2014





# Índice general

Índice de figuras	iii
Índice de tablas	v
Revisar	vii
Lista de símbolos y abreviaciones	ix
<b>1 Introducción</b>	<b>1</b>
1.1 Objetivos y estructura del documento . . . . .	3
<b>2 Marco teórico</b>	<b>5</b>
2.1 Espacios de Color . . . . .	5
2.1.1 Transformación RGB - YUV . . . . .	5
2.1.2 Transformación RGB - HSI . . . . .	6
2.2 Algoritmo de Canny . . . . .	6
2.2.1 Obtención del gradiente . . . . .	7
2.2.2 Supresión no máxima . . . . .	8
2.2.3 Histéresis de umbral . . . . .	8
2.3 Transformada de Hough . . . . .	8
2.3.1 Transformación lineal . . . . .	8
2.3.2 Transformación circular . . . . .	9
2.4 Procesador Blackfin BF609 . . . . .	10
2.4.1 Estructura . . . . .	11
<b>3 Solución propuesta</b>	<b>13</b>
<b>4 Resultados y análisis</b>	<b>15</b>
<b>5 Conclusiones</b>	<b>17</b>
Bibliografía	19
A Demostración del teorema de Nyquist	21
Índice alfabético	23



# Índice de figuras

2.1	Definición de la recta . . . . .	9
2.2	curvas senusoidales . . . . .	9
2.3	circulos . . . . .	10



# Índice de tablas



# Revisar

“RECORDARE” editar la acta .....	g
Se puede agregar imagenes del proceso del algoritmo.....	6
cambiar esta imagen por un gnuplot .....	9
insertar estas figuras .....	11





# Lista de símbolos y abreviaciones

## Abreviaciones

ASM	Modelos Activos de Forma
PCA	Análisis de componentes principales
WSN	Redes Inalámbricas de Sensores

## Notación general

<b>A</b>	Matriz.
$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$	
$\mathbb{C}$	Conjunto de los números complejos.
$\text{Im}(z)$ o $z_{\text{Im}}$	Parte imaginaria del número complejo $z$
$j$	$j = \sqrt{-1}$
$\text{Re}(z)$ o $z_{\text{Re}}$	Parte real del número complejo $z$
$\mathcal{T}[\cdot]$	Transformación realizada por un sistema
$\underline{\mathbf{x}}$	Vector.
$\underline{\mathbf{x}} = [x_1 \ x_2 \ \cdots \ x_n]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$	
$y$	Escalar.
$z^*$	Complejo conjugado de $z$



# Capítulo 1

## Introducción

En las últimas décadas se han desarrollado amplias investigaciones entorno al área de visión artificial (también conocida como visión por computador CV-Computer Vision)[1], con el fin de imitar las características del sistema visual humano tales como: acomodación, adaptación y discriminación ocular; y poder aplicarlas a actividades como: análisis de imágenes medicas, monitorización de señales de tránsito y rutas, vigilancia antirrobo y pruebas forenses, videojuegos interactivos y automatización de procesos industriales.

El área del sistema embebidos (Embedded Systems), también se han realizado grandes avances para el desarrollo de sistemas con una gran capacidad de procesamiento de información a un bajo costo, actualmente los sistemas embebidos están son omnipresentes en diversas aplicaciones como: la automatización industrial, en productos de consumo, comunicaciones y desarrollos militares. Por esto se estima que para el 2014, el mercado de la tecnología integrada ascenderá a 142,4 mil millones de dólares y se espera un crecimiento sostenido de un 7% anual por los próximos 5 años[3].

A finales de la década pasada (2009) nace el proyecto Alianza de Visión Embebida (EVA-Embedded Vision Alliance) con la idea de unificar estas dos tecnologías: visión artificial y sistemas embebidos. El objetivo principal de esta alianza es inspirar y empoderar a los diseñadores de sistemas embebidos y desarrolladores de aplicaciones a utilizar tecnología de visión integrada [4].

Debido al desarrollo de esta nueva tendencia de Visión Embebida, en el Instituto Tecnológico de Costa Rica, específicamente en el Laboratorio de Procesamiento de Señales e Imágenes (Sip-Lab), nace la iniciativa de incorporarse en esta nueva área, por esto se empiezan a adquirir nuevos equipos y tarjetas de desarrollo en Visión Embebida. El primer equipo en obtenerse fue el Blackfin Embedded Vision Starter Kit, que es una plataforma de hardware versátil, con las herramientas de desarrollo de software necesarias para permitir la construcción de alto rendimiento; está basado en un procesador Blackfin BF609 de bajo costo, el cual incluye un "procesador de canalización de visión" que simplifica y acelera los algoritmos de procesamiento de imágenes[?]. A pesar de toda la capacidad de esta plataforma, actualmente no se ofrecen datos reales sobre el rendimiento y velocidad

de procesamiento.

Un objetivo del Sip-Lab es, desarrollar aplicaciones en tiempo real con un bajo consumo energético, por esto es de vital importante para el laboratorio, tener información veraz sobre la capacidad de procesamiento y el consumo energético del Blackfin Embedded Vision Starter Kit ante diferentes escenarios en la utilización del hardware disponible en ella, pues esta información servirá como insumo a eventuales aplicaciones desarrollados por el laboratorio.

Este proyecto, viene a recopilar información sobre tiempo de comunicación entre núcleos y aceleradores gráficos, tiempos de ejecución y consumo energético de la arquitectura Blackfin BF609, esto con la ayuda de herramientas de desarrollo de sistemas embebidos como analizadores de protocolos, depuradores y emuladores, para categorizar posibles escenarios en el uso de los recursos.

La solución del problema, esta dividida en varias etapas, la primera es un algoritmo de una técnica de seguimiento de objetos por color, en la cual se realiza una discriminación de color estableciendo un rango umbral mediante un entrenamiento de colores, para esto existen varios métodos de segmentación de color, por ejemplo la segmentación basada en correspondencia con histograma en Hue sobre imagen HSL [?]; sin embargo para este proyecto se utiliza el método en planos de color HSV. Después de obtener la segmentación de color se realiza una validación geométrica basada en la método de Transformada de Hough para la detección de círculos. Para verificar la validez del algoritmo se implementa previamente con las bibliotecas de OpenCV[1].

Una vez validado el algoritmo de seguimiento por color se debe implementar en la plataforma Blackfin Embedded Vision Starter Kit, esto se hace definiendo diferentes escenarios con los recursos del procesador. Para esto se desarrollan dos propuestas: la primera consiste en la utilizar de un núcleo y acelerador de visión (pipeline vision processor) [7], en el núcleo se implementa el algoritmo de seguimiento por color y el despliegue de la interfaz gráfica, y en el acelerador de visión se realizan tareas de procesamiento de imágenes como filtrado y detección de bordes; en la segunda propuesta se utilizan los dos núcleos del procesador Blackfin BF609 [7] y el acelerador de visión, donde se separa el algoritmo y despliegue de la interfaz en núcleo diferentes.

Se pretende obtener los parámetros del rendimiento del procesador utilizando el dispositivo de depuración Emulator, ADZS-ICE-100B[6], esta herramienta permite realizar pruebas y depuración de aplicaciones en los procesadores de Analog Devices.

## 1.1 Objetivos y estructura del documento

Este proyecto tiene como objetivo diseñar una aplicación demostrativa para los investigadores del Laboratorio del procesamiento digital de señales de imágenes que informe sobre las características de rendimiento en la plataforma Blackfin Embedded Vision Starter Kit; que permita obtener las características de rendimientos, en forma de gráficos y tablas de tiempos de ejecución, latencias de sistemas de comunicación y consumo de energía. Para esto se pretende definir una técnica de seguimiento por color en tiempo real, que tenga una alta precisión con el mínimo costo computacional; validado por medio de la bibliotecas de OpenCV. Además se quiere implementar C/C++ al menos dos escenarios de uso del hardware del procesador Blackfin BF609; donde se obtenga la información sobre los recursos utilizados. Finalmente se pretende esquematizar el rendimiento de los escenarios en gráficos y tablas comparativas.

En el siguiente capítulo se esbozan los fundamentos teóricos necesarios para explicar en el capítulo 3 y en el capítulo 4 las propuestas realizadas. En el capítulo 5 se muestran algunos resultados experimentales de las características de rendimiento para los dos escenarios de utilización del hardware, así como su análisis a con referencia a un algoritmo de implementado con las bibliotecas de OpenCV. En el capítulo 6 se ofrecen las principales conclusiones, algunos aportes y recomendaciones para trabajos futuros.



# Capítulo 2

## Marco teórico

En este capítulo se presentan los conceptos que dan soporte a la solución implementada. Primero se hará una breve reseña sobre conceptos y algoritmos de procesamiento de imágenes: espacios de color, detector de borde Canny y transformación de Hough, luego se describe la composición de la arquitectura Blackfin.

### 2.1 Espacios de Color

Un espacio de color es una representación matemática tridimensional en la que se encuentran representados todos los colores visible para el ser humano, los más utilizados son el RGB y YUV. El espacio RGB se basa en los tres colores primarios aditivos rojo, verde y azul; es utilizado en el desarrollo de gráficos por computadora, por su parte el espacio YUV se define por una componente de luminancia (Y) y dos componentes de cromancia (UV); se usa principalmente en codificación de vídeo.

El HSI (matiz, saturación, intensidad) y HSV (matiz, saturación, valor), estos espacios de color se han desarrollado para ser más "intuitivo" en la manipulación de color y se han diseñado para aproximar la forma que los humanos perciben e interpretan color.

#### 2.1.1 Transformación RGB - YUV

La conversión del espacio RGB al YUV se define mediante el sistema

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (2.1)$$

En caso de la conversión YUV a RGB se tiene que

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 1,140 & 0 \\ 1 & -0,395 & -0,581 \\ 1 & 2.032 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (2.2)$$

La variables  $R', G', B'$  representa compensaciones de pantalla no lineal a partir de señales  $R, G, B$  puras[8].

### 2.1.2 Transformación RGB - HSI

La componente de tonalidad (H) de cada pixel RGB se obtiene mediante la ecuación:

$$H = \begin{cases} \theta & \text{si } B \leq G \\ 360^\circ - \theta & \text{si } B > G \end{cases} \quad (2.3)$$

con

$$\theta = \arccos \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right\} \quad (2.4)$$

La componente de saturación se obtiene por

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)] \quad (2.5)$$

El valor  $\min(R, G, B)$  representa el valor mínimo de arreglos de pixeles RGB.

Finalmente, la componente de intensidad esta dada por:

$$I = \frac{1}{3}(R + G + B) \quad (2.6)$$

## 2.2 Algoritmo de Canny

El algoritmo de Canny es una técnica de detección de bordes basada en la utilización la primera derivada, que toma valores de cero en regiones donde el cambio de intensidad no varía y tiene valor constante ante cualquier transformación de intensidad, por lo tanto los cambios abruptos en la intensidad son evidenciados por la primera derivada.

El algoritmo de Canny consiste en tres grandes pasos:

- Obtención del gradiente: se calcula la magnitud y orientación del vector gradiente en cada píxel.



- Supresión no máxima: se adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
- Histéresis de umbral: se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.

### 2.2.1 Obtención del gradiente

En este paso se aplica un filtro gaussiano para suavizar la imagen original y tratar de eliminar posible ruido existente, este filtrado se realiza promediando valores de intensidad de los píxeles en el entorno de vecindad con una máscara de convolución de media cero y desviación estándar  $\sigma$ .

$$M = \begin{bmatrix} 99 & 397 & 695 & 397 & 99 \\ 397 & 1986 & 3277 & 1986 & 397 \\ 695 & 3277 & 5362 & 3277 & 695 \\ 397 & 1986 & 3277 & 1986 & 397 \\ 99 & 397 & 695 & 397 & 99 \end{bmatrix} \quad (2.7)$$

Para la obtención del gradiente de intensidad se utiliza el operador de Sobel, con este operador es posible encontrar la magnitud del mayor cambio posible de intensidad y la dirección de este en cada pixel, al aplicar este operador se obtiene que tan bruscos o suave cambia la imagen en el punto analizado y consecuentemente que tan probable es encontrar un borde en la imagen y la orientación a la que tiende este.

La magnitud del gradiente de la intensidad esta dada por

$$M = \sqrt{s_x^2 + s_y^2} \quad (2.8)$$

y la dirección por

$$\theta = \arctan\left(\frac{s_y}{s_x}\right) \quad (2.9)$$

Los operadores de gradiente  $s_x$  y  $s_y$  se implementan mediante mascarar de convolución

$$s_x = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix} \quad s_y = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{bmatrix} \quad (2.10)$$

### 2.2.2 Supresión no máxima

La supresión no máxima es una técnica de adelgazamiento de bordes, que consisten en descartar los píxeles que no son máximos locales. Esto se realiza mediante el redondeo a la dirección del borde normal (vector gradiente) más cercano a  $45^\circ$  y la comparación de los píxeles de la dirección de borde normal positivo y negativo (es decir, si la dirección del borde normal es  $0^\circ$ , el píxel se compara con su vecino derecho e izquierdo), si el píxel no tiene mayor magnitud que sus vecinos, es suprimido.

### 2.2.3 Histéresis de umbral

Esta operación reduce los bordes falsos mediante el uso de dos umbrales: umbral bajo,  $T_L$ , umbral alto,  $T_H$ . Básicamente, la imagen es recorrida por píxel a píxel: si la intensidad del píxel es mayor que el umbral alto, entonces el píxel se considera fuerte

$$g_{NH}(x, y) = g_N(x, y) \geq T_H \quad (2.11)$$

si la intensidad es superior al umbral bajo, el píxel se considera débil.

$$g_{NL}(x, y) = g_N(x, y) \geq T_L \quad (2.12)$$

Finalmente si la intensidad del píxel es inferior al umbral bajo, el píxel es suprimido.

## 2.3 Transformada de Hough

La transformada de Hough es una técnica utilizada para aislar características de forma particular dentro de una imagen: identifica formas como líneas, círculos y elipses, es un método estándar en la detección de formas debido a su capacidad para hacer frente a los objetivos parcialmente ocluidos y su robustez a la presencia de ruido[5].

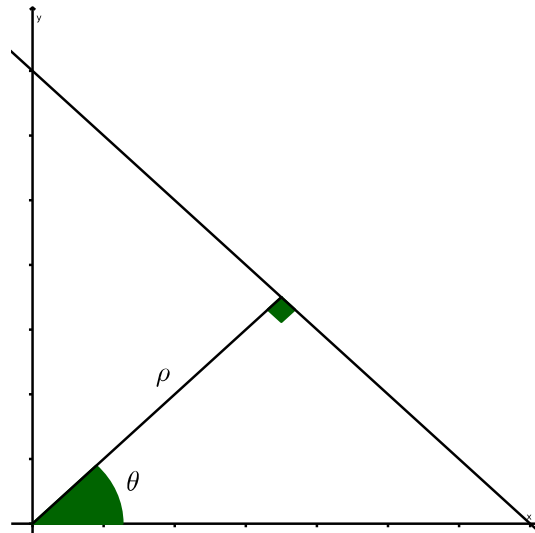
Este algoritmo utiliza la información de borde detectado para definir un mapeo de la orientación de un punto de borde a un punto de referencia de la forma. El punto de referencia puede ser pensado como el origen de una de coordenadas locales[2].

### 2.3.1 Transformación lineal

Una recta en el espacio de imagen, como la que se observa en la figura 2.1 se define como:

$$y = mx + n \quad (2.13)$$

donde  $m$  y  $n$  son pendiente y la intersección con el eje  $y$  respectivamente.



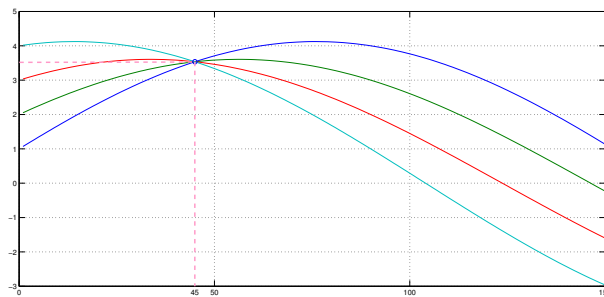
**Figura 2.1:** Definición de la recta

La idea fundamental de la transformación de Hough es obtener características en términos de  $m$ ,  $n$  y no considerar las coordenadas  $(x, y)$ , para esto la ecuación de la recta se debe mapear al espacio de Hough mediante la parametrización en términos del ángulo  $\theta$  y la distancia al origen  $\rho$ .

La ecuación de la recta en el espacio de Hough se puede escribir como:

$$\rho = x \cos \theta + y \sin \theta \quad (2.14)$$

y es representada mediante una curva sinusoidal para un punto  $(x_0, y_0)$ . Por esto, como se muestra en la figura 2.2 si dos curvas correspondientes en dos puntos se intersectan, existen una línea recta entre ellos en la imagen.



**Figura 2.2:** curvas senusoidales

↪ cambiar  
esta imagen  
por un  
gnuplot

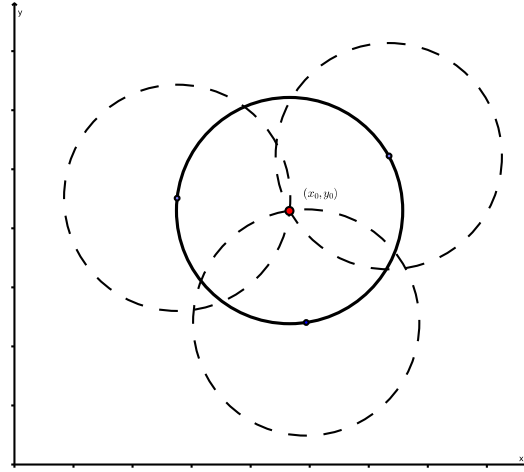
### 2.3.2 Transformación circular

En la transformación circular se determinan parámetros de un círculo a partir de ciertos números de puntos importantes en su perímetro. Un círculo de radio  $R$  y de centro  $(x_0, y_0)$

se describe por la ecuación

$$(x - x_0)^2 + (y - y_0)^2 = R^2 \quad (2.15)$$

Por lo tanto, el espacio de Hough para círculos es tridimensional, sin embargo si se define un valor  $R$  para todo círculo, entonces el espacio de Hough se reduce a un plano bidimensional donde un punto  $(x_0, y_0)$  es parte del perímetro de un círculo de centrado en  $(x, y)$  y además es compartido por otros dos círculos como en la figura 2.3, entonces este punto es el centro de un círculo existen en la imagen.



**Figura 2.3:** círculos

Si el valor de  $R$  es desconocido, entonces se realiza una proyección cada píxel en el espacio de Hough que forma un cono, pues si se establece un punto fijo en la ecuación del círculo se satisface un conjunto infinito de círculos concéntricos. El radio de los círculos centrado en los puntos en el perímetro del círculo aumenta a medida que la proyección de pixel crece, como se observa en la figura .... Si un número de píxeles que se encuentran en un círculo en el espacio de la imagen se proyecta en el espacio Hough, los conos resultantes se encuentran en puntos  $(x, y, r)$ . El algoritmo de acumulación de votos como se desarrolla en [9], producirá un máximo en  $(x, y, r)$  que se puede utilizar para reconstruir el círculo.

## 2.4 Procesador Blackfin BF609

El procesador Blackfin BF609 es una generación de 16 a 32 bits manufacturados por Analog Devices Inc; diseñada específicamente para satisfacer las demandas computacionales y las limitaciones de potencia de audio embebido, vídeo y aplicaciones de comunicaciones sobre la base de MSA desarrollada conjuntamente entre Analog Devices y Intel Corporation[?].

La arquitectura del procesador Blackfin BF609 se basa en una tubería de 10 etapas RISC MCU/DSP con un conjunto de instrucciones 16-/32-bit mixtas diseñadas para la densidad

de código óptimo. Esta arquitectura es compatible con el modelo SIMD[10] e incluye instrucciones para vídeo acelerado y procesamiento de imágenes.

↩ insertar  
estas figuras

Como se muestra en la Figura..., el procesador integra dos núcleos. Cada núcleo, se muestra en la Figura 2, contiene dos multiplicadores 16 bits, dos acumuladores de 40 bits, dos ALUs de 40 bits, cuatro ALUs para vídeo, y una unidad de desplazamiento de 40 bits. El proceso en las unidades de cálculo se realiza en datos de 8, 16 o 32 bits aportados por el arreglo de registro del procesador.

El archivo de registro contiene ocho registros de 32 bits, que también puede utilizarse como 16 registros independientes de bits. Todos los operandos para las operaciones de calculo provienen del archivo de registro multipuerto y campos constantes instrucciones. Las unidades MAC pueden realizar una de 16 bits al de 16 bits se multiplican en cada ciclo, la acumulación de los resultados en los acumuladores de 40 bits. La unidad de desplazamiento de 40 bits puede realizar desplazamientos, rotaciones y se utiliza para apoyar la normalización, extracciones de campos e instrucciones de campo de depósito. Las ALUs realizan el conjunto tradicional de operaciones aritméticas y lógicas en los datos de 16 bits o 32 bits. Además, ejecuta instrucciones especiales para acelerar diversas tareas de procesamiento de señales. Estas instrucciones especiales incluyen operaciones de bits como el extracto de campo y el recuento de la población, multiplicación de módulos, divisiones primitivas, la saturación y el redondeo, y la detección de la señal/exponente.

### 2.4.1 Estructura

El procesador utiliza un DMA para transferir datos dentro de los espacios de memoria o entre un espacio de memoria y algún periférico. El procesador puede especificar las operaciones de transferencia de datos y volver al proceso normal mientras el controlador DMA integrado plenamente realiza las transferencias de datos independientes de la actividad del procesador.

Se utilizan dos módulos de protección CRC [11] para la detección de errores en código y/o datos en la memoria, el en contenido de los registros mapeados por la memoria, o objetos de los mensajes de comunicación. También proporciona un sistema de control de eventos que soporta tanto la anidación y priorización.

La distribución de la memoria se realiza en único espacio de direcciones de 4 GBytes unificado, usando de direcciones de 32 bits. Todos los recursos del nucleos del procesador, incluida la memoria interna, memoria externa, y registros de control de E/S, ocupan secciones separadas de este espacio de direcciones común.

El procesador posee un subsistema de video conformado por un procesador de video, interfaz periférica paralela[] y un compositor de píxeles. En el procesador de video (PVP) se desarrollan algoritmos de procesamiento de señal e imágenes que se requieren para el preprocesamiento y coprocesamiento de video, se incluyen bloques para realizar convoluciones, conversión de coordenadas rectangular a polares, clasificación de pixeles por

diferenciación, operaciones aritméticas, segmentación por color[?] y escalamiento. La interfaz periférica paralela facilita la conexión directa a los paneles LCD, a los convertidores analógico a digital y digital a analógico, codificadores y decodificadores de vídeo, módulos de sensor de imagen y otros periféricos de uso general. Finalmente el compositor de píxeles (PIXC) proporciona superposiciones de imágenes con soporte transparente de color, mezcla alfa [alpha blending], y capacidades de conversión de espacio de color para la salida de las pantallas LCD y codificadores de vídeo NTSC/PAL.

# Capítulo 3

## Solución propuesta

Primero que todo, jamás utilice el título indicado arriba, sino algo relacionado con su solución: “Sistema de corrección de distorsión” o lo que competa a su tesis en particular.

Este capítulo puede separarse en varias secciones, dependiendo del problema concreto. Aquí los algoritmos o el diseño del sistema deben quedar lo suficientemente claros para que otra persona pueda re-implementar al sistema propuesto. Sin embargo, el enfoque no debe nunca concentrarse en los detalles de la implementación particular realizada, sino del diseño conceptual como tal.

Recuerdese que toda figura y tabla deben estar referenciadas en el texto.





## Capítulo 4

# Resultados y análisis

En tesis formales en este capítulo se exponen los diseños experimentales realizados para comprobar el funcionamiento correcto del sistema. Por ejemplo, si se realiza algún sistema con reconocimiento de patrones, usualmente esta sección involucra las llamadas *matrices de confusión* donde se compactan las estadísticas de reconocimiento alcanzadas. En circuitos de hardware, experimentos para determinar variaciones contra ruido, etc. También pueden ilustrarse algunos resultados concretos como ejemplo del funcionamiento de los algoritmos. Puede mostrar por medio de experimentos ventajas, desventajas, desempeño de su algoritmo, o comparaciones con otros algoritmos.

Recuerde que debe minimizar los “saltos” que el lector deba hacer en su documento. Por tanto, usualmente el análisis se coloca junto a tablas y figuras presentadas, y debe tener un orden de tal modo que se observe cómo los objetivos específicos y el objetivo general del proyecto se han cumplido.



# Capítulo 5

## Conclusiones

Las conclusiones no son un resumen de lo realizado sino a lo que ha llevado el desarrollo del proyecto, no perdiendo de vista los objetivos planteados desde el principio y los resultados obtenidos. En otras palabras, qué se concluye o a qué se ha llegado después de realizado el proyecto de graduación. Un error común es “concluir” aspectos que no se desarrollaron en la tesis, como observaciones o afirmaciones derivadas de la teoría directamente. Esto último debe evitarse.

Es usual concluir con lo que queda por hacer, o sugerencias para mejorar los resultados.



# Bibliografía

- [1] Open source computer vision, 2014. URL <http://opencv.org> [visitado el 11 de mayo de 2014]. 1, 2
- [2] D. H. Ballard. Readings in computer vision: Issues, problems, principles, and paradigms. chapter Generalizing the Hough Transform to Detect Arbitrary Shapes, pages 714–725. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. URL <http://dl.acm.org/citation.cfm?id=33517.33574>. 8
- [3] bccResearch. Embedded systems: Technologies and markets, 2014. URL <http://www.bccresearch.com> [visitado el 11 de mayo de 2014]. 1
- [4] John W. Eaton. Why does embedded vision need an alliance?, 2014. URL <http://www.embedded-vision.com> [visitado el 11 de mayo de 2014]. 1
- [5] J. Illingworth and J. Kittler. A survey of the hough transform. *Comput. Vision Graph. Image Process.*, 1988. URL [http://dx.doi.org/10.1016/S0734-189X\(88\)80033-1](http://dx.doi.org/10.1016/S0734-189X(88)80033-1). 8
- [6] Analog Devices Inc. Analog devices blackfin emulator, 2014. URL <http://www.analog.com> [visitado el 11 de mayo de 2014]. 2
- [7] Analog Devices Inc. Blackfin dual core embedded processor Hoja de datos, 2014. URL [http://www.analog.com/static/imported-files/data\\_sheets/ADSP-BF606\\_607\\_608\\_609.pdf](http://www.analog.com/static/imported-files/data_sheets/ADSP-BF606_607_608_609.pdf). 2
- [8] K. Jack. *Digital Video and DSP*, volume I. Elsevier Inc, 2008. 6
- [9] Carolyn Kimme, Dana Ballard, and Jack Sklansky. Finding circles by an array of accumulators. *Commun. ACM*, 18(2):120–122, February 1975. URL <http://doi.acm.org/10.1145/360666.360677>. 10
- [10] H. Lawson, B. Svensson, and L. Wanhammar. *Parallel Processing in Industrial Real Time Applications*. Prentice Hall, 1992. 11
- [11] M. Waqar, T. Thanh Hoang, and P Edefors. Cyclic redundancy checking (crc) accelerator for the flexcore processor. 2010. URL <http://www.cse.chalmers.se/~hoangt/papers/tunghoang-dsd2010.pdf>. 11



# Apéndice A

## Demostración del teorema de Nyquist

El título anterior es solo un ejemplo ilustrativo. Éste teorema no ameritaría un apéndice pues es parte normal del currículum de Electrónica, pero apéndices usualmente involucran aspectos de esta índole, que se salen de la línea de la tesis, pero que es conveniente incluir por completitud.

Los anexos contienen toda información adicional que se considere pertinente agregar, como manuales de usuario, demostraciones matemáticas que se salen de la línea principal de la tesis, pero que pueden considerarse parte de los resultados del trabajo.





# Índice alfabético

objetivos, 3