

TESIS DEFENDIDA POR

Martín Mancilla Gómez

Y aprobada por el siguiente comité:

---

Dr. Gustavo Olague Caballero

*Director del Comité*

---

M.C. José Luis Briseño Cervantes

*Miembro del Comité*

---

Dr. Andrei Tchernykh Graboskaya

*Miembro del Comité*

---

Dr. Eugenio Rafael Méndez Méndez

*Miembro del Comité*

---

Dr. Benjamín Hernández Valencia

*Miembro del Comité*

---

Dr. Hugo Homero Hidalgo Silva

*Coordinador del Programa en  
Ciencias de la Computación*

---

Dr. David Hilario Covarrubias Rosales

*Director de Estudios  
de Posgrado*

24 de Septiembre del 2010

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN  
SUPERIOR DE ENSENADA



---

POSGRADO EN CIENCIAS  
EN CIENCIAS DE LA COMPUTACIÓN

---

**Reconocimiento de Objetos con Características Locales  
Inspiradas en la Corteza Visual a través de Programación  
Genética**

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN CIENCIAS

Presenta:

**Martín Mancilla Gómez**

Ensenada, Baja California, México. Septiembre del 2010.

**RESUMEN** de la tesis de **Martín Mancilla Gómez**, presentada como requisito parcial para obtener el grado de MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN. Ensenada, Baja California. Septiembre del 2010.

## **Reconocimiento de Objetos con Características Locales Inspiradas en la Corteza Visual a través de Programación Genética**

Resumen aprobado por:

---

Dr. Gustavo Olague Caballero

*Director de Tesis*

En la última década, una parte significativa de la comunidad científica en visión por computadora ha adoptado diversos enfoques basados en la detección y descripción de rasgos locales para resolver una amplia variedad de problemas; como lo son: el reconocimiento y detección de objetos, problemas de indexación de imágenes, recuperación de imágenes en base a su contenido, así como la clasificación de múltiples clases de objetos. En particular, el problema del reconocimiento de varias clases de objetos considerando imágenes naturales ha demostrado ser un reto difícil para la visión por computadora. De esta forma, la mayoría de los sistemas de clasificación de múltiples objetos están manualmente optimizados para trabajar con un tipo particular de escena. Sin embargo, el sistema visual humano es capaz de reconocer cualquier objeto siguiendo el mismo criterio a través del flujo de información visual que se lleva a cabo en la ruta ventral. En esta tesis la meta principal es proponer un enfoque basado en el paradigma del cómputo evolutivo que nos permita estudiar y sintetizar operadores a través de programación genética (GP), mediante un modelo jerárquico inspirado en las capas V1, V2 y V4 de la corteza visual humana y en el funcionamiento de las células simples y complejas. Esto facilita la extracción de características locales en imágenes digitales pertenecientes a una misma clase de objetos; iniciando en la corteza visual primaria V1 y terminando en la corteza infero temporal donde se lleva a cabo el reconocimiento de los objetos. Esta última parte se simula a través de la máquina de vector de soporte (SVM). Los resultados que se muestran en este trabajo ofrecen tres tipos de contribuciones. Primero, una implementación de un modelo estándar inspirado en la ruta ventral o ruta del ¿qué? Segundo, la aceleración del modelo por medio de la unidad de procesamiento gráfico (GPUs), que aprovecha las capacidades de cálculo en paralelo. Y por último, un enfoque novedoso en la extracción de características locales por medio de programación genética. Estos resultados experimentales confirman las

conclusiones de nuestro trabajo y demuestran que los resultados son comparables en calidad pero con una eficiencia mayor respecto a aquellos del estado-del-arte.

**Palabras clave:** corteza visual, ruta ventral, reconocimiento de objetos, clasificación de clase de objetos, programación genética, GP, máquina de vector de soporte.

**ABSTRACT** of the thesis presented by **Martín Mancilla Gómez**, as a partial requirement to obtain the MASTER SCIENCE degree in COMPUTER SCIENCES. Ensenada, Baja California. august 2010.

## **Object Recognition with Local Features Inspired by the Visual Cortex through Genetic Programming**

Abstract approved by:

---

Dr. Gustavo Olague Caballero

*Thesis director*

In the last decade, a significant part of the scientific community in computer vision has taken several approaches based on the detection and description of local features to solve a wide variety of problems; such as: object recognition and detection, image indexing, content based image retrieval and multi-class object recognition. In particular, the problem of recognition of various classes of objects considering natural images has proved to be a difficult challenge for computer vision. Thus, most classification systems of multiple objects are manually optimized to work with a particular type of scene. However, the human visual system can recognize any object following a single criterion through the flow of visual information occurring in the ventral stream. In this thesis, the main goal is to propose an approach based on the paradigm of evolutionary computation that allows us to study and synthesize operators through genetic programming (GP), using a hierarchical model inspired from the layers V1, V2 and V4 of the human visual cortex and operation of simple and complex cells. This facilitates the local feature extraction in digital images within the same class of objects, starting in the primary visual cortex V1 and ending on the infero temporal cortex which performs the recognition of objects. This last part is simulated through a support vector machine (SVM). The results shown in this work offer three types of contributions. First, an implementation of a standard model inspired by the ventral stream or sometimes called the “what pathway”. Second, the acceleration of the model using the graphics processing unit (GPUs), that leverages the capabilities of parallel computing. Finally, a novel approach to local feature extraction using genetic programming. These experimental results confirm the conclusions of our work and show that the results are comparable in quality but with a higher efficiency with respect to those of the state-of-the-art.

**Keywords:** visual cortex, the ventral stream, object recognition, object class recognition, genetic programming, GP, SVM.

*Dedicado a mis padres*

**Bertha Gómez Hernández**  
**y**

**Martín Mancilla Murillo**

# Agradecimientos

Agradezco a mis padres, a mi hermana Magaly y mi hija Danna Kamila por su apoyo incondicional sin el cual no hubiera podido lograr este sueño.

Al Dr. Gustavo Olague, por haberme aceptado en su grupo de investigación EvoVisión. Por las grandes conversaciones y las numerosas discusiones filosóficas científicas que me han hecho crecer como persona.

A mis sinodales por sus aportaciones y observaciones a este trabajo.

A mi esposa Katty que me brindó su comprensión en este camino, que sin su cariño y ánimo hubiera sido interminable.

A mis amigos y compañeros de *EvoVisión*, de los cuales siempre tuve su apoyo y amistad.

A mis amigos de la *Krusti Cueva* por hacer mi estancia más agradable, así como a mis amigos que siempre me apoyaron y me dieron ánimos en tiempos difíciles.

Al Centro de Investigación Científica y de Educación Superior de Ensenada y al Consejo Nacional de Ciencia y Tecnología, por darme las facilidades para estudiar en ésta institución de alto prestigio internacional.

Ensenada, México  
24 de Septiembre del 2010.

Martín Mancilla Gómez

# Tabla de Contenido

Capítulo	Página
<b>Resumen</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>Agradecimientos</b>	<b>vi</b>
<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tablas</b>	<b>xv</b>
<b>I Introducción</b>	<b>2</b>
I.1 Descripción del problema . . . . .	4
I.2 Investigación previa relevante . . . . .	12
I.3 Objetivos del trabajo . . . . .	18
I.3.1 Objetivos Específicos . . . . .	18
I.4 Importancia de la investigación . . . . .	19
I.5 Organización de la tesis . . . . .	20
<b>II La Corteza Visual</b>	<b>21</b>
II.1 Análisis de la forma visual . . . . .	24
II.2 La corteza cerebral . . . . .	25
II.3 Estructura histológica de la corteza visual primaria . . . . .	28
II.3.1 Estría de Gennari-Vick d'Azir en la capa IV . . . . .	28
II.3.2 Tipos celulares en la corteza visual . . . . .	29
II.3.3 Estratificación de la corteza visual . . . . .	30
II.3.4 Circuito de retroalimentación en V1 . . . . .	31
II.4 Campos receptores en la corteza visual y detección de contornos . . . . .	32
II.4.1 Tipos neuronales en V1 según su campo receptor . . . . .	32
II.4.2 Tipos neuronales según su campo receptor en la corteza circunstriada . . . . .	37
II.4.3 Hipótesis propuestas sobre las conexiones entre las células de la vía visual . . . . .	40
II.5 Organización modular de la corteza visual, Percepción de la forma y movimiento . . . . .	46
II.5.1 Organización modular (columnar) en la corteza visual primaria (V1) . . . . .	46
II.5.2 Columnas de orientación . . . . .	46
II.5.3 Columnas de dominancia ocular . . . . .	48
II.5.4 Gotas o burbujas . . . . .	49
II.5.5 Segregación funcional en V1 . . . . .	51
II.6 Corteza visual circunscrita o de asociación (áreas visuales de asociación)	52
II.6.1 División funcional de la corteza preestriada o circunstriada . . . . .	53
II.6.2 Organización por columnas y segregación funcional en V2 . . . . .	55

# Tabla de Contenido (Continuación)

Capítulo	Página
II.6.3	Análisis de la forma dinámica en V3 . . . . . 55
II.6.4	Análisis cromático en V4 . . . . . 56
II.6.5	Análisis del movimiento en V5 (MT) . . . . . 56
II.7	Corteza temporal inferior (íferotemporal) . . . . . 58
II.8	Corteza parietal posterior . . . . . 60
II.9	Integración final de la información visual . . . . . 60
<b>III</b>	<b>Programación Genética</b> . . . . . <b>64</b>
III.1	Fundamentos de la Programación Genética . . . . . 66
III.1.1	Conceptos Básicos . . . . . 68
III.2	Terminales y Funciones . . . . . 69
III.2.1	Conjunto de Terminales . . . . . 69
III.2.2	Conjunto de Funciones . . . . . 70
III.2.3	Estructura de un programa a través de programación genética . 71
III.3	Inicialización de la Población . . . . . 75
III.3.1	Inicialización de las estructuras con árboles . . . . . 76
III.3.2	El método de mitad y mitad . . . . . 77
III.4	Operadores Genéticos . . . . . 78
III.4.1	Cruza . . . . . 78
III.4.2	Mutación . . . . . 80
III.5	Aptitud y Selección . . . . . 81
III.5.1	Función de Aptitud . . . . . 81
III.5.2	Algoritmo de Selección . . . . . 82
III.5.3	Algoritmo Básico de GP . . . . . 83
III.6	Problemas resueltos exitosamente por la GP . . . . . 85
<b>IV</b>	<b>GPGPU</b> . . . . . <b>87</b>
IV.1	GPU vs. CPU . . . . . 88
IV.2	CUDA . . . . . 91
IV.2.1	Arquitectura CUDA . . . . . 92
IV.2.2	Estructura de un programa en CUDA . . . . . 98
IV.2.3	Threads, blocks y grids . . . . . 99
IV.2.4	Concurrencia CPU-GPU . . . . . 102
<b>V</b>	<b>Reconocimiento de objetos</b> . . . . . <b>104</b>
V.1	Descripción y extracción de características . . . . . 107
V.1.1	Requisitos en la extracción de características . . . . . 108
V.2	Clasificación . . . . . 109
V.2.1	Clasificación supervisada . . . . . 109
V.2.2	Clasificación no supervisada . . . . . 121

# Tabla de Contenido (Continuación)

Capítulo	Página
<b>VI Reconocimiento de objetos inspirado en la corteza visual</b>	<b>123</b>
VI.1 Implementación del modelo HMAX . . . . .	125
VI.1.1 Capa de filtros de Gabor - S1 . . . . .	127
VI.1.2 Capa de invarianza local - C1 . . . . .	129
VI.1.3 Capa de características locales - S2 . . . . .	131
VI.1.4 Capa de invarianza global - C2 . . . . .	133
VI.1.5 Experimentos . . . . .	134
<b>VII Modelo HMAX con Programación Genética</b>	<b>143</b>
VII.0.6 Detalles de Implementación . . . . .	145
VII.0.7 Herramientas utilizadas . . . . .	150
<b>VIII Conclusiones, aportaciones y trabajo futuro</b>	<b>152</b>
<b>BIBLIOGRAFIA</b>	<b>155</b>

# Lista de Figuras

Figura		Página
1	Trayectorias visuales en el cerebro. . . . .	6
2	Lóbulos de la corteza cerebral. . . . .	7
3	Trayectorias visuales secundarias en donde se muestran la ruta del ¿Dónde? o Dorsal y la del ¿Qué? o Ventral. En esta figura se muestra que la ruta del donde me dice donde se encuentra el objeto y la ruta del que, me dice que tipo de objetos es. . . . .	8
4	Áreas visuales corticales. La corteza visual primaria (V1) recibe información directamente del núcleo lateral geniculado. . . . .	9
5	Trayectorias que toman la información a partir de la corteza visual primaria (V1) hacia la corteza parietal y corteza infero temporal. . . . .	10
6	Distintos kernels de Gabor en las cuatro orientaciones (0°, 45°, 90° y 135°). . . . .	13
7	Esquema gráfico del modelo jerárquico. Cada nivel recibe las entradas de un mayor número de unidades “células”. . . . .	14
8	Cartografía de las diversas áreas corticales según Korbinian Brodmann (1909). La imagen del lado izquierdo está subdividida de acuerdo a las superficies lateral de un hemisferio cerebral y la segunda imagen, está basado en la superficie media o interna-superficial . . . . .	27
9	Estratificación de la corteza visual. Se representan las conexiones de los sistemas parvocelulares (P), magnocelular (M) y koniocelular (I) del cuerpo geniculado lateral (CGL), así como las conexiones verticales entre las propias capas del córtex y las eferencias que desde éste van a otras zonas cerebrales. . . . .	30
10	Respuestas de una célula simple a una barra con diversas orientaciones (de Hubel y Wiesel , 1959). . . . .	34
11	Tipos de campo receptor simple (de Hubel y Wiesel , 1959). . . . .	35
12	Células complejas con respuesta selectiva a la dirección del movimiento (de Hubel y Wiesel , 1962). . . . .	36
13	a) Respuestas de una célula compleja a diversa longitudes de un estímulo en forma de barra. La respuesta se va incrementando mientras la longitud de la barra aumenta aproximadamente 2° de arco. Más allá la respuesta no varía. b) Respuesta de una célula compleja “con inhibición terminal”. Una vez que se supera en 2° la ampliación de la longitud de la línea, la respuesta decae (adaptado de Hubel , 1982) . . . . .	38

## Lista de Figuras (Continuación)

Figura	Página
14 Cuando se estimula la región activadora central de su campo receptor, esta célula compleja con inhibición terminal total responde vigorosamente. Si el estímulo es más alargado e incide ampliamente en una de las regiones inhibitorias, la respuesta decae ostensiblemente. Cuando se cambia la orientación del estímulo y se hace incidir en una de las regiones inhibitorias, apenas existe inhibición, lo que demuestra que la orientación óptima para el efecto inhibitorio es la misma que para la región central (adaptado de Hubel , 1982). . . . .	39
15 Explicación de cómo a partir de varios campos receptores de células estrelladas, puede formarse un campo receptor alargado correspondiente a una célula simple. a) Convergencia de cuatro neuronas estrelladas (capa IV) en una simple. b) Estímulo en barra con orientación óptima, que da la máxima respuesta. c) Estímulo con orientación ortogonal al anterior, que da respuesta nula (adaptado de Hubel y Wiesel , 1962). . .	41
16 a) Convergencia de tres células corticales simples en una compleja. El campo receptor de ésta no discriminará exactamente una posición dentro de su campo receptor relativamente extenso. Si el estímulo sale de él habrá respuesta inhibitoria (adaptado de Hubel y Wiesel , 1962). b) Por otro lado, al ir excitando sucesivamente células simples, detectará movimiento, de forma diferente si lo hace de izquierda a derecha que de derecha a izquierda, ya que activará en un orden diferente las células simples (Barlow y Levick , 1965). . . . .	42
17 Modelos alternativos para explicar el campo receptor de una célula compleja con “inhibición terminal”. a) La célula inhibitoria “B”, cubre enteramente el campo receptor. La célula inhibitoria apenas responde cuando se estimula la región a mediante una línea corta, pero lo hace muy vigorosamente, si se sobrepasa esta región en uno o ambos extremos mediante una línea más larga. b) Convergencia de tres células complejas ordinarias en una de “inhibición terminal”. La célula que determina la región central del campo receptor es excitatorio. (A), mientras que las dos que determinan las regiones adyacentes, son inhibitorias (B y C) (adaptada de Hubel , 1982). . . . .	43
18 a) Es estímulo óptimo para esta célula es un ángulo en movimiento y orientado ( $90^\circ$ ), que no invada la zona inhibitoria de la derecha (segundo registro) (adaptado de HuWe:1965). b) Línea curva: estímulo óptimo para una célula que tenga su campo receptor conformado por una zona excitatoria flanqueada por dos inhibitorias (adaptado de Hubel , 1982). . .	44

# Lista de Figuras (Continuación)

Figura		Página
19	Dibujo de un gato durmiendo realizado abstrayendo 38 puntos de máxima curvatura y uniéndolos mediante líneas rectas (adaptado de Attneave , 1954). . . . .	45
20	Ejemplo de la reconstrucción de un registro electrográfico cortical en el gato. En la porción superior de la figuram, el electrodo sigue aproximadamente un columna de orientación. Una vez que atraviesa la sustancia blanca y debido al plegamiento de la corteza, atraviesa sucesivamente de manera oblicua varias columnas celulares que responden a orientaciones diferentes (Hubel y Wiesel , 1962). . . . .	47
21	Esquema de una hipercolumna o módulo que incluye las “burbujas”. Se muestran asimismo las diferentes proyecciones de las células del CGL (adaptado de Livinstone y Hubel , 1984). . . . .	50
22	Relaciones de la corteza visual primaria con otras áreas (inferotemporal, circunscrita y campo ocular frontal) en el macaco. . . . .	53
23	Localización del área visual primaria (V1) y de otras áreas del procesamiento de la información visual (V2, V3, V3A, V4 y V5) en el córtex visual del macaco (adaptado de Zeki y Shipp , 1988). . . . .	54
24	Módulo cortical en el área V5 (adaptado de Rodamn <i>et al.</i> , 1989). . . . .	58
25	Experimento de Gross (Gross <i>et al.</i> , 1972). El número indica la intensidad de la respuesta en células del córtex inferotemporal de un macaco: 1 no evoca respuesta; 2-3 Ligero incremento; 4-5 respuesta gradual cada vez más intensa; 5 representa el perfil de una mano humana; y 6 respuesta con máxima intensidad al perfil de una mano de macaco de su misma especie. . . . .	59
26	Vías parvo y magnocelulares desde la retina y CGL a través de V1 y V2 hasta las áreas V3, V4, V5 e IT. (adaptación a partir de los datos de Livinstone y Hubel , 1984; Zeki y Shipp , 1988) . . . . .	63
27	Rama de un árbol. . . . .	70
28	Estructura de un árbol. . . . .	72
29	Estructura Lineal. . . . .	74
30	Grafo acíclico dirigido. . . . .	75
31	Individuo inicial utilizando con el método “grow” y con una PMP=4. . . . .	76
32	Individuo inicial utilizando el método “full” y PMP=3. . . . .	78
33	Máxima Profundidad de 3, inicializando con el método “full”. . . . .	79
34	Mutación a un individuo. . . . .	80
35	Comparación entre incremento de velocidad de los CPU’s y los GPU’s . . . . .	89
36	Diferencias entre la CPU y la GPU. . . . .	90

# Lista de Figuras (Continuación)

Figura		Página
37	Nvidia CUDA . . . . .	92
38	Esquema del hardware de una GPU. . . . .	95
39	Esquema de una GPU cuando ejecuta aplicaciones en CUDA. . . . .	96
40	Estructura de la memoria en la GPU. . . . .	98
41	Estructura de la memoria en la GPU. . . . .	99
42	Estructura de la ejecución de un programa en CUDA. . . . .	100
43	Threads, bloques y grids. . . . .	101
44	Ejemplo orden de ejecución de los threads de un bloque. . . . .	102
45	Etapas que conforman la visión por computadora. . . . .	105
46	Ejemplo del algoritmo $K - nn$ . Se desea clasificar el círculo verde. Para $k=3$ este es clasificado con la <b>clase triángulo</b> , ya que existe solo un cuadrado y 2 triángulos, dentro del círculo que los contiene. Si $k=5$ este es clasificado con la <b>clase cuadrado</b> , ya que hay 2 triángulos y 3 cuadrados, dentro del círculo externo. . . . .	113
47	Separación de un conjunto de datos mediante SVM . . . . .	120
48	Esquema gráfico del modelo jerárquico. Cada nivel recibe las entradas de un mayor número de unidades llamadas “celulas”. El diccionario de características está compuesto de 200 parches con su cuatro orientaciones de Gabor ( $0^\circ$ , $45^\circ$ , $90^\circ$ y $135^\circ$ ) . . . . .	126
49	Ejemplo de filtros de Gabor de una escala dada para las 4 orientaciones usadas ( $0^\circ$ , $45^\circ$ , $90^\circ$ y $135^\circ$ ). . . . .	127
50	Piramide de imágenes a 12 escalas diferentes. . . . .	129
51	Imagen convolucionada con los filtros de Gabor en sus diferentes orientaciones. . . . .	130
52	Por cada escala se generan 4 imágenes convolucionadas con filtros de gabor, generando un total de 48 imágenes. . . . .	131
53	En la Operación Max el número 1 se realiza primero entre las escalas . . . . .	132
54	En la Operación MAX el número 2 se generan a partir de la imagen entre 2 escalas generando un total de 6 bandas con un total de 24 imágenes . . . . .	133
55	Recapitulando la capa C1. . . . .	134
56	Selección de los parches. . . . .	135
57	Creación del diccionarios de características (features) . . . . .	136
58	Operaciones que se realiza por cada parche en las Capas S2. . . . .	137
59	Operacion de MAX . . . . .	138
60	Operacion de MAX . . . . .	140
61	Selección de las características de las clases de objetos . . . . .	141

## Lista de Figuras (Continuación)

Figura		Página
62	Matriz de Confusión 76% de clasificación en imágenes de pruebas. Se clasificaron correctamente 114 de 150 imágenes . . . . .	141
63	Matriz de Confusión 71.33% de clasificación en imágenes de pruebas. Se clasificaron correctamente 107 de 150 imágenes . . . . .	142
64	Tiempo que toma el modelo HMAX en imágenes de diferentes tamaños. . . . .	142
65	Modelo estándar de la ruta ventral con programación genética. . . . .	146
66	Proceso de evolución de los operadores de características. . . . .	147
67	Etapa de evaluación del modelo estándar con GP. . . . .	147
68	Ejemplo de operadores matemáticos evolucionados y aplicados a una imagen. . . . .	150

# Lista de Tablas

Tabla		Página
I	Parámetros del GP utilizados en el HMAX . . . . .	148
II	Clases de objetos, se utilizan 30 ó 15 imágenes de forma aleatoria de las 100 imágenes por clase . . . . .	149



# Capítulo I

## Introducción

La inteligencia Artificial (AI, Artificial Intelligence) es el campo de la investigación científica que intenta imitar la inteligencia humana, concretamente intenta resolver problemas de interpretación, aprendizaje y razonamiento cognitivo, a través de la creación de herramientas para hacer cálculos, comparaciones y razonamiento con lógica formal, algoritmos genéticos, entre algunos otros. Al mismo tiempo que los biólogos empezaban a entender cómo es que los seres vivos vemos, otro grupo de científicos e ingenieros, especialistas en computación e inteligencia artificial, comenzaban a explorar la idea de la visión artificial, es decir; que las computadoras pudieran analizar e interpretar las imágenes como lo hacemos los humanos, naciendo una nueva disciplina denominada Visión por Computadora.

Sin duda el planteamiento original más elaborado sobre cuáles son los problemas que tiene que resolver la visión, es el realizado por David Marr en su libro Vision (Marr, 1982). Marr deja como preguntas abiertas lo siguiente:

- ¿Qué están haciendo realmente las áreas cerebrales de la corteza cerebral?
- ¿Cuáles son los problemas que al definir los algoritmos requieren explicación?
- ¿Cuál es el nivel de descripción en el que deben buscarse tales explicaciones?

El enfoque computacional propuesto por Marr, entre otros, establece que el análisis de la visión se centra en tres aspectos fundamentales:

1. Definir cuáles son los problemas en la percepción visual.

2. Proponer soluciones a estos problemas.
3. y entender como es que estos procesos se ejecutan en el sustrato subyacente (biológico/neuronal o artificial).

En otras palabras, el problema de visión se define como una serie de tareas que se deben resolver proponiendo soluciones algorítmicas a estos problemas, e intentar explicar como es que esos procesos se llevan a cabo en los sistemas biológicos reales.

Para Marr la aproximación de la fisiología parecía más prometedora. El desarrollo de la tecnología permitía desde los años 40 el registro de la actividad de una única neurona en animales vivos. Esto lleva al descubrimiento de los “campos receptivos” de las neuronas (Hartline , 1940) y a la caracterización de la complejidad a la que una única neurona es capaz de responder en niveles cada vez más profundos de la trayectoria visual (Hubel y Wiesel , 1962; Hubel y Wiesel , 1968). La investigación realizada en este subcampo ha influido grandemente en el estudio del funcionamiento del sistema visual humano, pues de manera análoga se podrá cumplir el propósito principal de la visión por computadora. Por otro lado, nuestro cerebro le da una gran importancia a la información que nos llega a través de la vista, y el área del cerebro dedicada a este sentido es la mayor con respecto a la dedicada a los otros sentidos. El ser humano percibe a las imágenes a través de la retina del ojo, como si fuera una clase de cámara fotográfica, en la que incide la luz y permite el comienzo del proceso de visión, es decir, este proceso nos permite ver nuestro entorno. De esta manera, unos minúsculos receptores, llamados conos y bastones, perciben la intensidad de luz en los rasgos del color rojo hasta el violeta (llamada luz visible). Los bastones son más sensibles a la intensidad lumínica y nos permite ver cuando hay muy poca luz. Los conos perciben el color, algunos son más sensibles al rojo en un 65%, otros al verde en 33% y otros al azul en tan sólo el 2% (Gonzalez y Woods , 2001); y mediante la combinación de estos

tres colores, llamados básicos o primarios, podemos percibir toda una gama de colores. Una vez que está información es recolectada por la retina es enviada al cerebro por una serie de fibras, que conforman el llamado nervio óptico (Ng *et al.* , 2007). Pero surgen las siguientes preguntas;

- ¿Qué sucede después?
- ¿Cómo es que podemos reconocer los diferentes objetos que percibimos?
- ¿Cómo sabemos al ver a ciertas personas que es ella y no otra persona?
- ¿De qué forma determinamos la posición en el espacio de un objeto, i.e., como tomar un objeto e identificarlo?

Bajo estas mismas preguntas se creó la Visión por Computadora, considerada como un conjunto de técnicas y modelos que permiten el procesamiento, análisis e interpretación de la información considerada relevante en una escena. La Visión por Computadora tiene como objetivo principal: Proveer del sentido de la vista a sistemas autónomos para que puedan interactuar de forma eficiente en ambientes complejos (Branch y Olague , 2001). Así mismo, la investigación realizada en esta rama de la inteligencia artificial, ha influido decisivamente en el estudio del funcionamiento del sistema visual humano, pues de manera análoga se podrá cumplir el propósito principal de la visión por computadora.

## I.1 Descripción del problema

En la actualidad, la mayoría de los sistemas de clasificación de objetos en imágenes ó reconocimiento de clase de objetos, están manualmente optimizados para trabajar con una clase particular de objetos, por ejemplos caras (Liu , 1995; Schneiderman y Kanade , 2000; Viola y Jones , 2001), o automoviles (Schneiderman y Kanade , 2000). Por otro

lado, el ser humano es capaz de reconocer cualquier objeto siguiendo un mismo criterio, independientemente de la clase de objeto que sea. Recientemente, han aparecido sistemas capaces de manejar satisfactoriamente cualquier clase de objeto siguiendo una metodología común (Agarwal y Roth , 2002; Fei-Fei *et al.* , 2004; Fergus *et al.* , 2003; Serre *et al.* , 2005b).

Así, podemos abordar el problema de la clasificación con un enfoque basado en la figura del objeto en su conjunto(Liu , 1995; Schneiderman y Kanade , 2000) o entendiendo al objeto como un conjunto de partes que siguen una determinada relación espacial (Agarwal y Roth , 2002; Fei-Fei *et al.* , 2004; Mohan *et al.* , 2001), o sin dicha relación (Serre *et al.* , 2004). Durante el proceso de entrenamiento del clasificador, en un gran número de los algoritmos propuestos hasta ahora, se usan datos de entrenamiento segmentados y alineados para su correcto funcionamiento (Liu , 1995; Schneiderman y Kanade , 2000), además de un número grande de muestras de entrenamiento (Viola y Jones , 2001. Esto último ocurre principalmente en aquéllos que extraen información basada en características locales, Si queremos emular el Sistema Visual Humano (*SVH*) debemos encontrar procedimientos que no contemplen estos inconvenientes. Los modelos propuestos por Poggio y Serre (Serre *et al.* , 2004) y Fei-Fei (Fei-Fei *et al.* , 2004) son capaces de aprender a partir de un pequeño conjunto de muestras de entrenamiento no segmentadas para cualquier tipo de clase de objeto.

Para reconocer cualquier objeto se propone tomar como referencia el sistema visual humano, intentando emular el proceso que este lleva a cabo. Recientemente, mediante experimentos con animales, en particular primates, hemos empezado a adentrarnos al misterio de cómo se procesan e interpretan las imágenes en nuestro cerebro. Con base en estos descubrimientos, que iniciaron con las investigaciones de Hubel y Weisel

(Hubel y Wiesel , 1959), sabemos que la información que proviene de la retina se divide en dos partes, una se dirige ó fluye al hemisferio cerebral izquierdo y otra al derecho (Ng *et al.* , 2007). Se ha descubierto también que no hay una relación única entre un ojo y un hemisferio! Aproximadamente la mitad de la información de los ojos va a un hemisferio y la otra mitad al otro. En la Figura 1 ilustra cómo la información percibida en la retina de ambos ojos viaja mediante el nervio óptico hasta llegar a la corteza visual, donde se procesa la información.

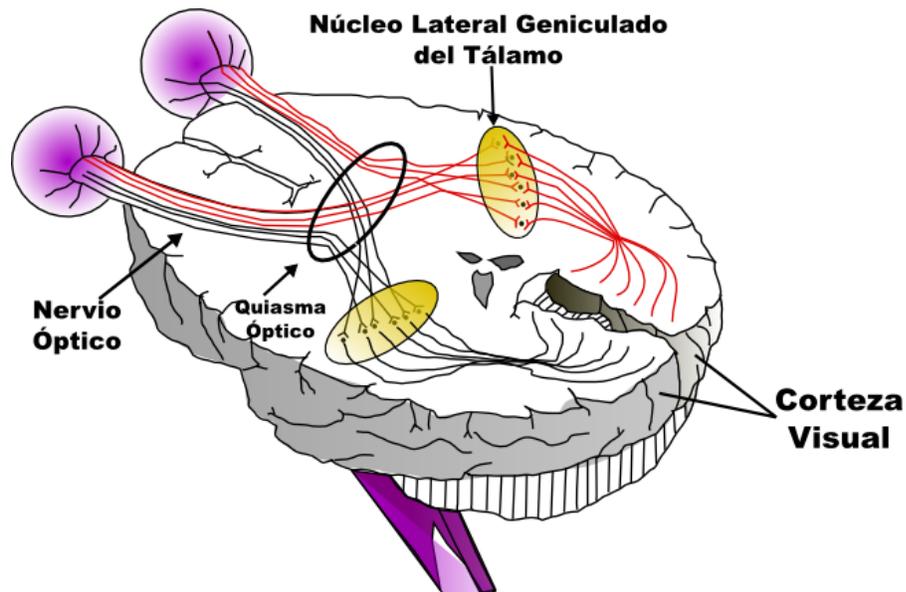


Figura 1: Trayectorias visuales en el cerebro.

La información visual se transporta mediante el nervio óptico a la corteza cerebral, a una zona que está ubicada en la parte posterior de la cabeza, llamada corteza visual. Ahí inicia el procesamiento de la información visual en el cerebro (Cajal , 1899).

Dentro de la corteza cerebral se encuentra la zona más importante donde inicia el procesamiento de la información capturada a través de los ojos que es la **corteza visual**

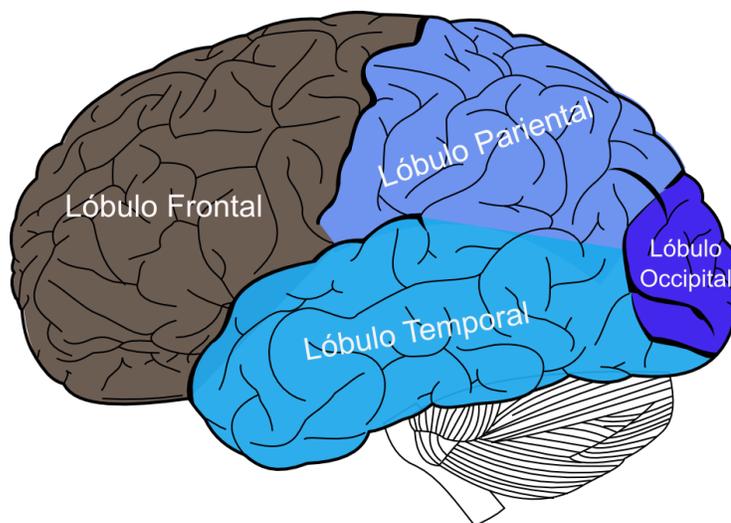


Figura 2: Lóbulos de la corteza cerebral.

**primaria**, conocida como “*corteza estriada*” o V1. Está recibe información visual que parte de una estación de relevo en el tálamo. En los humanos la corteza visual primaria está en la superficie media caudal de los hemisferios cerebrales, extendiéndose ligeramente sobre el polo hemisférico posterior en el lóbulo occipital (Ver Figura 2). En estas áreas se codifican características visuales como color, luminancia, frecuencia espacial, orientación y movimiento. A partir de esas propiedades extraídas de los objetos visibles el cerebro construye la realidad visual. Para ello el cerebro se organiza en dos flujos principales de procesamiento de la información:

1. La información visual que viene del mundo exterior es procesada por las células de la retina (conos y bastones) y se transmite por el nervio óptico hacia el núcleo geniculado lateral del tálamo, y de ahí va a la capa V1 (polo caudal, lóbulo occipitales). Esta trayectoria se conoce como *retino-geniculado-estriado* o trayectoria visual primaria (ver Figura 1) y es la que tiene más importancia en el momento de la construcción de nuestra representación visual del mundo en el cerebro.
2. La trayectoria visual secundaria está constituida por dos vías que parten del

lóbulo occipital, en la región de la corteza visual primaria o V1, la primer vía se dirige hacia el lóbulo temporal y la segunda se dirige hacia el lóbulo parietal (Ver Figura 3). Estas dos vías participa en funciones de localización y reconocimiento de objetos.

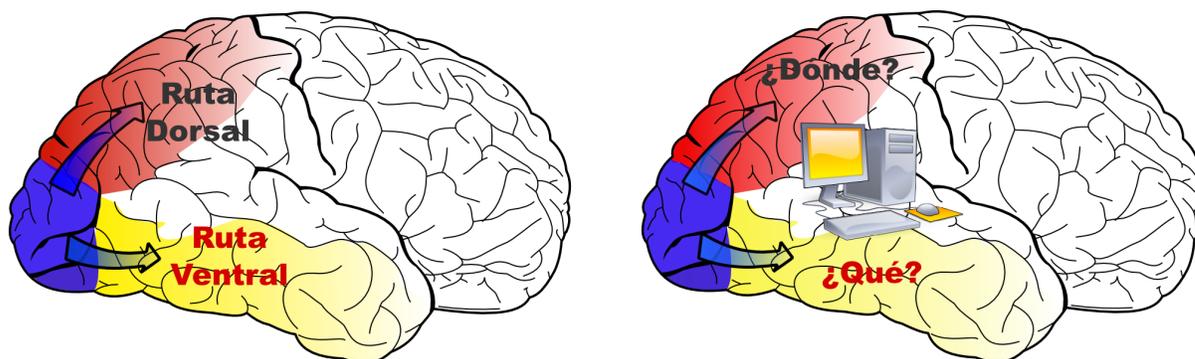


Figura 3: Trayectorias visuales secundarias en donde se muestran la ruta del ¿Dónde? o Dorsal y la del ¿Qué? o Ventral. En está figura se muestra que la ruta del donde me dice donde se encuentra el objeto y la ruta del que, me dice que tipo de objetos es.

Alrededor de la corteza visual también conocida como corteza estriada o V1, existe una región visual cortical grande llamada “**corteza visual extraestriado**” también nombradas como capas V2, V3, V4, V5, V6 (lóbulo parietal y temporal) (ver Figura 4). Existen más de 36 áreas visuales distintas en el cortex extraestriado de los primates. Estas áreas visuales contienen mapas parciales redundantes del mundo visual, es decir que, la misma información se representa en distintos sectores corticales, pero cada uno se especializa en el análisis de ciertos aspectos específicos que componen una escena. Por ejemplo, en una escena hay información sobre formas, color, movimiento y localización. Así, la información de esos tipos se representa en distintas áreas existiendo un área específica que se especializa en procesar cada clase de información sobre el color,

forma, etc.

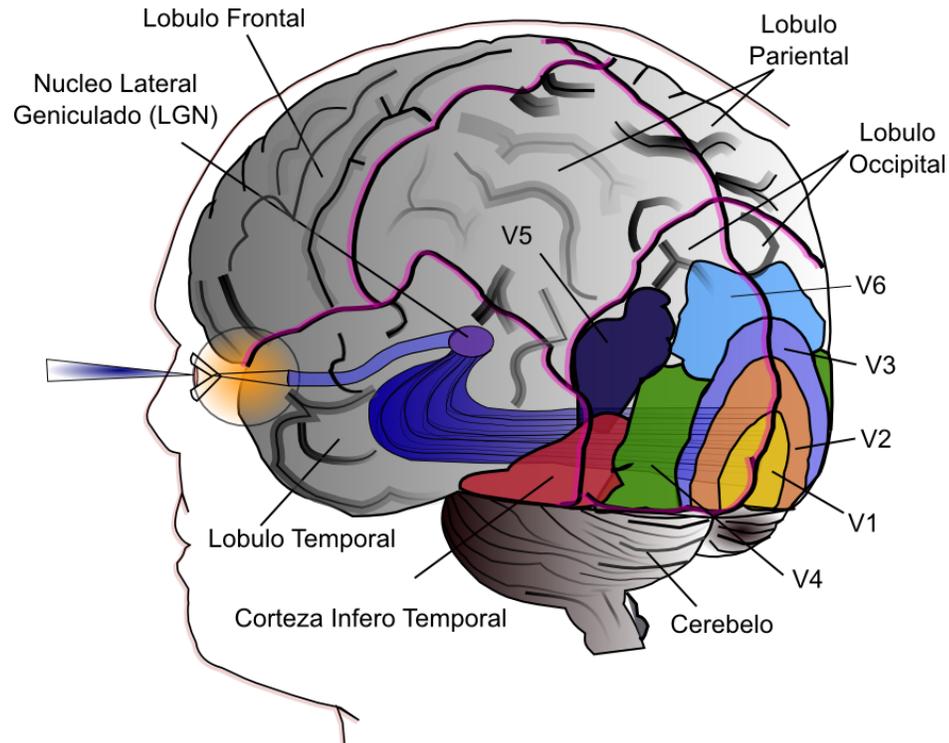


Figura 4: Áreas visuales corticales. La corteza visual primaria (V1) recibe información directamente del núcleo lateral geniculado.

Como ya mencionamos, la trayectoria visual secundaria se encuentra dividida en 2 vías (ver Figura 3), que inician desde la corteza estriada hasta el extraestriado. Éstas sustentan flujos de información característicos, cuya descripción es la siguiente:

1. La ruta “**Ventral**”, conocida en la literatura como la trayectoria del **¿Que?**, está trayectoria se encarga de la extracción de la información relacionada con la identidad de los objetos. Está fluye desde V1 hacia el lóbulo temporal. Comienza específicamente en V1 cruza a través del área visual V2, luego a través del área

visual V4 y finalmente llega a la corteza Infero Temporal. Ésta ruta se encuentra asociada a la representación y reconocimiento de los objetos.

2. La ruta “**Dorsal**”, conocida también como la trayectoria del **¿Dónde?**. Está se proyecta desde V1 hacia el lóbulo parietal, es decir; comienza en V1 cruza a través del área visual V2, luego al área visual medio temporal (*MT*) o también conocida como V5. Esta ruta lleva información sobre movimiento del estímulo del objeto y se localización dentro del espacio visual. Durante este flujo de información se determinan localizaciones que serán susceptibles de ser el objetivo de una acción inminente.

También existe una interconectividad muy densa dentro de y hacia las trayectorias dorsal y ventral, ver Figura 5.

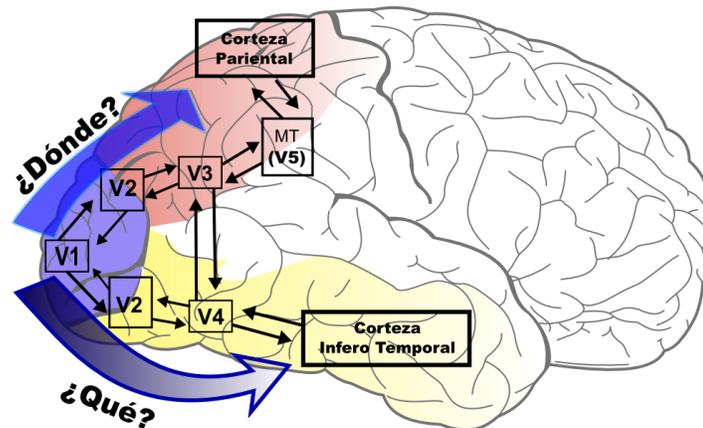


Figura 5: Trayectorias que toman la información a partir de la corteza visual primaria (V1) hacia la corteza parietal y corteza infero temporal.

Cómo se ha mencionado en éste capítulo, para llevar a cabo el reconocimiento de

objetos se utilizarán la ruta ventral teniendo como trayectoria: V1- V2 - V4 - IT<sup>1</sup>. Bajo este procesamiento que se lleva a cabo en la corteza visual para el reconocimiento de objetos, surgen modelos inspirados biológicamente los cuales imitan ciertas características del procesamiento visual humano. Sin embargo en ésta investigación se utilizará el modelo computacional para la clasificación de clase de objetos que se encuentra basado en la propuesta de T. Serre (Serre *et al.* , 2005b), el cual utiliza una arquitectura llamada HMAX (Hierarchical Model and X) de reconocimiento de objetos que combina tolerancia a la posición y su escala en base a detectores de bordes de varios tamaños inspirados por la corteza visual de mamíferos como el gato y el primate. El modelo está formado por una jerarquía de cuatro niveles que corresponden a la organización observada en células visuales del área visual V1 a V4 del la corteza visual. El modelo se encuentra formado por representaciones de cuatro capas de células nerviosas llamadas S1, C1, S2 y C2 que corresponden fisiológicamente a la ruta que toman los impulsos visuales desde la corteza visual primaria, a la corteza infero temporal y finalmente a la corteza pre-frontal (ver Figura 5). Cada capa toma como entrada la salida de la capa anterior y crea jerárquicamente información más compleja sobre el objeto observado, de tal forma que es posible reconocer el objeto.

Por otro lado, los sistemas de reconocimiento de objetos dentro de la visión por computadora poseen tres partes bien definidas: la etapa de procesamiento, en la que se extrae el objeto; la etapa del esquema de representación, la cuál se hace una descripción de la características relevantes del objeto y la etapa de estrategia de reconocimiento, consistirá en la clasificación de los objetos presentes en la imagen (Roy *et al.* , 2004). En la primera y segunda etapa se utilizará un modelo estandar inspirado de la ruta ventral y en la etapa de reconocimiento de clases de objetos se apoya de la máquina

---

<sup>1</sup>Infero Temporal

de vector de soporte para su correcta clasificación.

## I.2 Investigación previa relevante

En éste trabajo, la investigación relevante gira alrededor de 3 temáticas: la primera basada en el modelo computacional para el reconocimiento de objetos inspirado en la corteza visual; la segunda en el cómputo evolutivo para resolver problemas de visión, utilizando la programación genética (GP<sup>2</sup>); y finalmente utilizamos los conceptos de programación masivamente paralela a través del uso de la tarjeta gráfica (GPGPU<sup>3</sup>) para acelerar los algoritmos.

En este paradigma de la corteza visual, el trabajo más sobresalientes en el que se basará la síntesis del modelo HMAX (Hierarchical Model and X), propuesto por T. Serre (Serre *et al.* , 2005a). Éste modelo computacional simula la ***ruta ventral*** de la corteza visual (ver Figura 7), a continuación se describe brevemente el modelo:

**S1:** La primera capa, se integra por un conjunto conocido de células simples, llamadas S1, las cuales están formadas por neuronas ubicadas en V1. Las células de S1 tienen la característica de responder solo si en determinada región de su campo visual existe una línea recta alineada con su orientación preferida, a éstas características se les conoce como reaccion celular que responde a la preferencia de barras de determinada orientación. Esta reacción celular es bien conocida desde el trabajo de Hubel y Wiesel (Hubel y Wiesel , 1959; Hubel y Wiesel , 1962; Hubel y Wiesel , 1965) Existen distintas células sensibles a segmentos de recta de 0°, 45°, 90°y 135°. Estas orientaciones son suficientes y se cree que no restringen

---

<sup>2</sup>GP: por sus siglas en inglés “*Genetic Programming*”

<sup>3</sup>GPGPU: por sus siglas en inglés “*General-Purpose Computing on Graphics Processing Units*”

de manera significativa la capacidad de reconocimiento (Riesenhuber y Poggio, 1999). Las células de S1, como ya se ha mencionado responde a a diferentes orientaciones, es decir tienen campos visuales variables; para realizar está representación se codificó en su implementación con filtros de Gabor (ver Figura 6) utilizando un ancho variable en el número de píxeles.

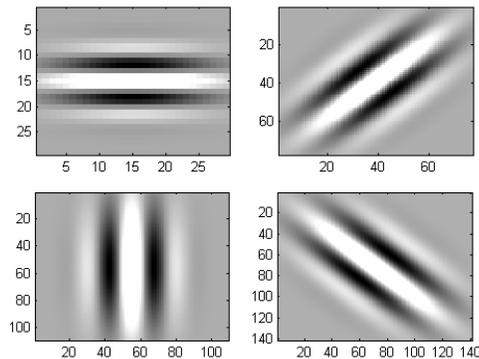


Figura 6: Distintos kernels de Gabor en las cuatro orientaciones ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ ).

**C1:** Las células del nivel siguiente, o unidades “complejas” conocidas como C1 se integran por una cantidad menor en comparación de las células S1, esto se debe a que reciben la salida de múltiples células S1. Es por esto que las células C1 tienen un campo visual mayor y son más específicas a detalles complejos. Las operaciones que realizan estas células sobre las respuestas de S1 consisten, de acuerdo al modelo, en una operación de máximo de los valores de gris de las celdas tomadas en los mapas generados por S1. El ancho de las celdas de C1 es un indicativo de que tanta información poseen dichas unidades respecto a su tamaño. Nótese que aún se conservan las orientaciones como mapas independientes, mostrados en la Figura 6.

**S2:** En el siguiente nivel S2 o nivel simple 2, se simula un campo de visión doble al

de cada célula de nivel anterior, por lo que dichas células que contienen mayor información ya que se integran nuevamente varias células simples.

**C2:** La siguiente operación es la de C2 que toma por cada escala los cuatro tipos de orientaciones y obtiene el máximo por cada respuesta de S2. En otras palabras, se toma la célula más activa de las cuatro orientaciones. Con esto se pueden obtener los bordes de los objetos para su respectiva clasificación y categorización, utilizando el método de clasificación conocido como máquina de vector de soporte.

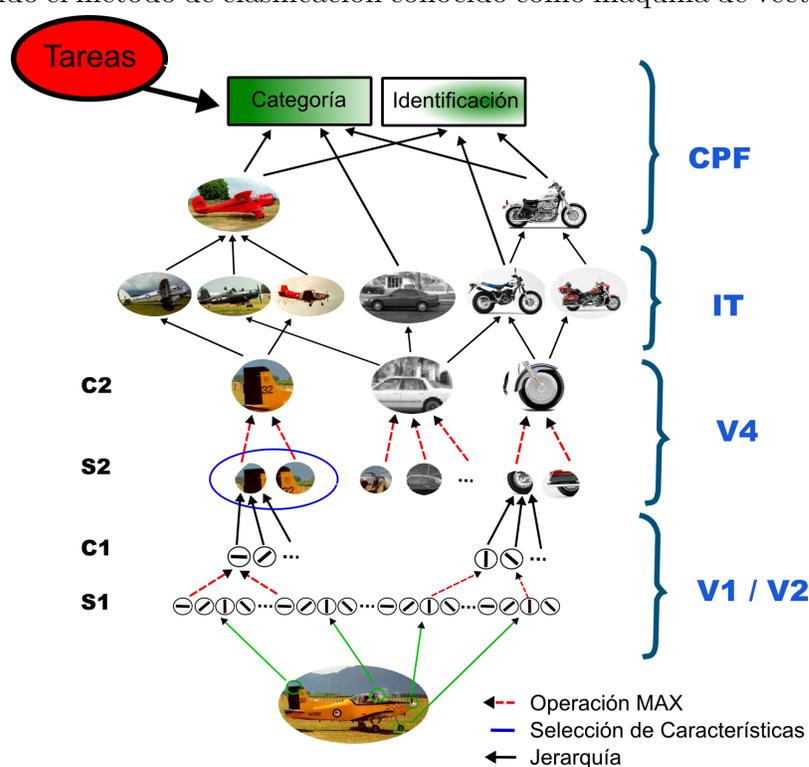


Figura 7: Esquema gráfico del modelo jerárquico. Cada nivel recibe las entradas de un mayor número de unidades “células”.

La segunda temática bajo la cuál se desarrollará este trabajo, es mediante técnicas evolutivas, en particular utilizaremos las técnicas conocidas como visión por computadora evolutiva. Esta área se refiere a la generación automática de sistemas de visión por computadora a través de técnicas de cómputo evolutivo (Cagnoni *et al.* ,

2008). Los primeros trabajos en visión por computadora evolutiva iniciaron en los 90's. Lohmann, un pionero de este campo, mostro la idea de cómo utilizar estrategias evolutivas (Lohmann , 1991). Sin embargo, al inicio no se sabía claramente cuál era la principal técnica que podría crear exitosamente operadores adaptativos que se aproximarán al óptimo (Olague , 2007). Ahora es claro que la GP puede resolver particularmente el análisis de una imagen. Hoy en día, la visión por computadora evolutiva a llegado a investigar áreas muy activas dentro de la visión por computadora, cómo lo son el análisis de textura, descriptores y reconocimiento de características en una escena, entre otras, pero no aquellas relacionadas al sistema visul humano.

La GP es una técnica de tipo evolutiva, basada en la metodología inspirada por la evolución biologica para hallar programas de cómputo que realicen una tarea específica predefinida por el usuario. Está es una técnica del aprendizaje de maquina, usada para optimizar una población de programas de computadora generados de acuerdo a una función de aptitud la cual sirve para evaluar que tan bueno es cada programa (individuo) en su desempeño de realizar la tarea dada.

La GP evoluciona programas de computadora representados principalmente como estructuras de árboles. Los árboles pueden ser facilmente evaluados de forma recursiva. Cada nodo del árbol tiene una función operador, y cada nodo terminal tiene un operando, con esto se forma una representación matemática fácil de evaluar y evolucionar. Así la PG evoluciona los programas con dos operaciones fundamentales, conocidas como la cruza y la mutación. Dichas operaciones simulan los procesos evolutivos de intercambio de información.

Dentro de la visión por computadora evolutiva se encuentra la metodología que se

tomará como referencia para esta tesis, la cual se explica en el trabajo desarrollado por Trujillo y Olague, en este trabajo de investigación se describe el algoritmo GP propuesto para sintetizar, de forma automática, operadores empleados durante el proceso de detección de puntos de interés (Trujillo y Olague, 2006), dentro del espacio de búsqueda. A continuación se explica esta metodología. El algoritmo consiste en definir un conjunto de Funciones y Terminales que se utilizan para definir el espacio de búsqueda:

$$F = \{+, | + |, -, | - |, I_{out}, *, \div, I_{out}^2, \sqrt{I_{out}}, \log_2(I_{out}), EQ(I_{out}, k \cdot I_{out})\}$$

$$U\left\{\frac{\delta}{\delta_x}G_{\sigma D}, \frac{\delta}{\delta_y}G_{\sigma D}, G_{\sigma=1}, G_{\sigma=2}\right\}$$

$$T = \{I, L_x, L_{xx}, L_{xy}, L_{yy}, L_y\}$$

donde  $I$  es la imagen de entrada, y  $I_{out}$  puede ser cualquier terminal en  $T$ , o la salida de una de las funciones de  $F$ ;  $EQ(I)$  es una normalización de histograma;  $L_u$  representan derivadas Gaussianas en la dirección  $u$  de la imagen;  $G_\sigma$  son filtros de suavizado Gaussiano;  $\frac{\delta}{\delta_u}G_{\sigma D}$  son derivadas de la función Gaussiana; y por último se define la constante  $k = 0.05$ . En el caso de la función  $\log_2$  se toma la convención de  $\log_2(0) = 0$  para evitar operaciones con cantidades infinitas. Por razones similares, cuando se da el caso en que se aplica la raíz a un valor negativo  $\sqrt{-x}$ , solo se toma la parte real del resultado para evitar operaciones posteriores con número complejos.

La tercera temática es la concerniente al uso de la GPGPU. Éste es un concepto reciente dentro de la computación, y se refiere a la utilización de las Unidades de Procesamiento Gráfico ( $GPU$ ), que se encarga de aprovechar las capacidades de cómputo de una GPU. La tarjeta gráfica es un procesador diseñado para los cálculos implicados

en la generación de gráficos. El alto poder de cómputo y la flexibilidad introducida en las recientes GPU's están permitiendo solucionar problemas, hasta hace poco desafiantes para las computadoras, usando hardware de bajo costo. Esto ha generado cambios importantes en todo el contexto de procesamiento paralelo ya que ha puesto al alcance el hardware masivamente paralelo de un alto rendimiento que puede incluso ser utilizado en resolución de problemas no necesariamente del dominio gráfico (Fung *et al.* , 2005). Esta tecnología se consideran atractivas para su uso en aplicaciones especialmente en el ámbito científico y de simulación. Por ejemplo, se han desarrollado técnicas para la implementación de simulaciones de fluidos, bases de datos, algoritmos de clustering, la aceleración de algoritmos de visión por computadora, entre algunos otros.

En el campo de visión por computadora y bajo esta nueva tendencia en el mundo de la programación GPGPU se está aprovechando la GPU. Podemos mencionar los siguientes trabajos importantes como lo son; detección de objetos (Fung *et al.* , 2005); puntos de interés en una imagen en tiempo real (Sinha *et al.* , 2006); aceleración de algoritmos en visión por computadora evolutivos sobre imágenes (Harding y Banzhaf , 2008); reconocimiento de objeto similares dentro de una escena en tiempo real (Ebner , 2009). El ambiente de desarrollo es CUDA<sup>4</sup>, que es un conjunto de herramientas que explota las ventajas de las GPU's frente a las CPUs de propósito general utilizando el paralelismo que ofrecen sus múltiples núcleos, que permiten el lanzamiento de un altísimo número de hilos simultáneamente. Por lo tanto, si una aplicación está diseñada para realizar tareas independientes que requiere un alto costo de calculo computacional (CUDA , 2007).

---

<sup>4</sup>CUDA: del inglés Compute Unified Device Architecture

## I.3 Objetivos del trabajo

El objetivo general de este trabajo es: sintetizar la obtención de características locales de la arquitectura HMAX (Hierarchical Model and X) que combinen tolerancia a transformaciones en base a detectores de características útiles en el reconocimiento de objetos. Esta investigación está inspirada en la corteza visual a través del flujo de información visual de la ruta ventral, la cuál se analiza bajo la metodología de programación genética.

### I.3.1 Objetivos Específicos

Para alcanzar este objetivo, se han identificado varias etapas intermedias, las cuales se presentan como objetivos específicos que, al alcanzarse gradualmente ayudarán al cumplimiento del objetivo general:

- Analizar la arquitectura del modelo estándar de la ruta ventral, que lleva a cabo el reconocimiento de objetos inspirándose en el funcionamiento del sistema visual humano y su corteza visual.
- Implementar y evaluar dicho modelo en Matlab.
- Acelerar el modelo estándar de la ruta ventral usando la tecnología GPGPU (General-Purpose Computing on Graphics Processing Units).
- Utilizar la metodología de PG para evolucionar la extracción de características de rasgos sobre las imágenes pertenecientes a clases de objetos.
- Evaluar el modelo de la ruta estándar de la ruta ventral mediante PG para obtener el reconocimiento de clases de objetos

## I.4 Importancia de la investigación

En éste trabajo de investigación los resultados son satisfactorios, ya que se está automatizando las tareas de reconocimiento de clases de objetos. Esto coloca a nuestra metodología dentro del dominio del aprendizaje máquina. Además, al estar respaldado por una metodología comprensible y que ha presentado buenos resultados como lo es la programación genética, se asegura su usabilidad, entendimiento y la facilidad de ser modificado para satisfacer diferentes objetivos dentro del reconocimiento de objetos.

Además, la implementación de un sistema novedoso con el uso de un conjunto de métodos ya existentes (programación genética, reconocimiento de objetos motivado por un sistema biológico) que, en su conjunto, no han sido utilizados anteriormente.

La mayoría de los métodos de reconocimiento publicados, trabajan bien con imágenes seleccionadas cuidadosamente para sus experimentos. En este trabajo, se evaluará el algoritmo implementado utilizando un conjunto de datos:

- Un conjunto muy pequeño de imágenes de entrenamiento, manejando entre 15 y 30 por cada clase de objeto.
- Con variaciones significativas dentro del mismo conjunto de datos.
- Con presencia de oclusión parcial y variaciones de apariencia debido a cambios de ángulo de vista, iluminación y escala, y
- Un conjunto considerable de clases de objetos para su correcta clasificación.

## I.5 Organización de la tesis

Esta tesis está estructurada de la siguiente manera:

- **Capítulo II.** Se introducen los conceptos básicos de de la corteza visual, acompañados de antecedentes fisiológicos en el área, necesarios para el entendimiento de la investigación aquí descrita.
- **Capítulo III.** Se presentan los conceptos de programación genética y . se explica con detalle los fundamentos del cómputo evolutivo.
- **Capítulo IV.** Se muestra la tecnología GPGPU para aprovechar el paralelismo de la tarjeta gráfica.
- **Capítulo V.** Se introduce al lector en el tema del reconocimiento de objetos dentro de la visión por computadora, se tratan conceptos básicos importantes para el entendimiento de la metodología.
- **Capítulo VI** Se describen la implementación del modelo estandar inspirado en la ruta ventral de la corteza visual, realizados durante la investigación. Así como los resultados obtenidos durante las pruebas.
- **Capítulo VII.** Se da una explicación de la metodología implementada bajo programación genética para el modelo estandar de la ruta ventral para llevar a cabo el reconocimiento de objetos.
- **Capítulo VIII.** Finalmente se plantea la discusión de conclusiones, aportaciones y trabajo futuro sobre esta investigación.

## Capítulo II

# La Corteza Visual

El estudio anatómico-funcional de la corteza cerebral implicada en procesos visuales ha sido uno de los campos de la neurociencia que más ha avanzado en las últimas dos décadas. Puede decirse, sin temor a parecer exagerado, que un gran conocimiento de lo que actualmente sabemos sobre la integración cortical de la información sensorial proviene de estudios realizados en el sistema visual, de manera particular, en relación a las funciones intracorticales y subcorticales de la corteza cerebral encargada del procesamiento visual. El objetivo de este capítulo, es poner al alcance del lector algunos hechos neurobiológicos experimentales de relevancia, que han permitido avanzar mucho en los últimos años. En particular aquellos referidos a la anatomía funcional de la corteza cerebral relacionada con la visión. Los trabajos de Davis Hubel y Torsten Wiesel supusieron una auténtica revolución en el conocimiento debido a la profundidad en la explicación de cómo se lleva a cabo el procesamiento de la información visual por la corteza cerebral, desde el punto de vista morfofuncional. Además, estos estudios representaron un avance extraordinario en el conocimiento de la organización funcional de la corteza cerebral. En 1981 estos neurocientíficos recibieron el Premio Nobel de Medicina y Fisiología por sus trabajos sobre el procesamiento cortical de los impulsos visuales y sobre el procesamiento cortical de la información visual.

La corteza visual primaria o área 17 de Broadmann se sitúa en la cara medial del lóbulo occipital y se extiende hacia la convexidad en el polo occipital. De forma macroscópica, la característica principal de esa corteza es la presencia de la cisura

calcarina, que recorre de anterior a posterior esta zona de la corteza cerebral y es la responsable del abultamiento que se produce en el ventrículo lateral, denominado clásicamente con el término de *calcar avis* (Cisura calcarina), la cual presenta dos labios: el superior o dorsal y el inferior o ventral. Dicha región es una corteza granular que muestra una reducción de las capas piramidales y un marcado desarrollo de la capa IV (granular), y que, a su vez, se ha subdividido en varias subcapas. Así, la subcapa IVb de baja densidad celular corresponde a la estría de Gennari en las tinciones mielínicas y se caracteriza también por presentar las células granulares gigantes de Meynert. Por el contrario, las otras dos subcapas (IVa y IVc) presentan gran cantidad de células, con neuronas granulares más pequeñas y son dos de los lugares de la corteza cerebral donde existe una mayor celularidad. Desde el punto de vista morfofuncional, es importante señalar que la capa IVc puede subdividirse, a su vez, en dos porciones: IVc alfa y IVc beta (Kandel *et al.* , 1995). Así mismo, la corteza visual primaria tiene también una capa VI bastante desarrollada.

Alrededor de la corteza visual primaria se encuentran las áreas de Broadmann 18 y 19, clásicamente adscritas como cortezas visuales secundarias, las cuales están íntimamente relacionadas con el área 17 y que, como veremos más adelante, son muy importantes para el procesamiento posterior de la información visual a nivel cortical. Desde el punto de vista histológico, el área 18 presenta una capa IV bastante homogénea con neuronas granulares y el área 19 es una corteza de transición con las cortezas asociativas temporal y parietal. Existen, además, numerosas áreas corticales en los lóbulos temporal y parietal que se encargan de procesar gran cantidad de información visual y que detallaremos más adelante. En la actualidad, se tiende a señalar a las cortezas visuales extraestriadas con la denominación V2, V3, V4, etc. De acuerdo con la mayor o menor lejanía respecto a la corteza visual primaria o estriada (V1) y

que no se ajustan exactamente a las subdivisiones en áreas realizadas por Broadmann.

Además de la disposición altamente mielinizada de la capa IV de la corteza estriada existe otra distinción morfoquímica muy importante entre la corteza visual primaria y las cortezas vecinas. De esta manera, el área 17 de Broadmann presenta unas estructuras globulares en las capas supragranulares e infragranulares que son positivas para la enzima cicromoxidasa y se relacionan muy directamente con la percepción del color (Hubel y Wiesel , 1962). En las cortezas vecinas, dichas estructuras globulares desaparecen como tales y así encontramos otras formaciones positivas para dicha enzima las cuales forman a modo de bandas nuevas estructuras más alargadas gruesas y finas (Hubel , 1988).

En el inicio de la vía visual, los impulsos luminosos estimulan los fotorreceptores hiperpolarizando su potencial de membrana, lo que permite que la información nerviosa alcance, tras varias sinapsis, las células ganglionares de la retina. Finalmente, estas descargan potenciales de acción y sus axones inician una complicada ruta anatómica para alcanzar la corteza cerebral de forma más directa tras una estación talámica en el cuerpo geniculado lateral (vía visual geniculada), o de forma más compleja a través del colículo superior y del tálamo (vía visual extrageniculada). La disposición particular de las distintas estructuras que componen el globo ocular hace que la imagen visual incida sobre la retina de fomar real e invertida. Así, el cruce de fibras retinianas en el quiasma óptico permite que, tras la estación talámica correspondiente, se represente en la corteza visual primaria el campo visual contralateral con una inversión dorsoventral en la cisura calcarina de las porciones superiores e inferiores del mismo.

Cabe destacar que los trabajos de Hubel y Wiesel empezaron como la continuación

de los trabajos realizados por Kuffler en las células ganglionares de la retina. Kuffler había descrito la presencia de células con campos receptivos circulares con respuesta excitatoria (*on*) o inhibitoria (*off*) en el centro del campo receptivo, la cual produce una excitación o inhibición en la periferia del mismo, respectivamente (Kuffler , 1953). Con toda lógica, la exploración de la respuesta a estímulos luminosos de las células en la vía visual debería extenderse también al cuerpo geniculado lateral y a la corteza cerebral, concretamente la corteza visual primaria: este es el trabajo que empezaron a realizar Hubel y Wiesel a finales de los años 50 (Hubel y Wiesel , 1959; Hubel y Wiesel , 1962).

## II.1 Análisis de la forma visual

El primer nivel de análisis de la imagen en la retina sería a modo de mosaico, es decir, a partir de una gran cantidad de elementos discretos. Los medios dióptricos del ojo proyectan una imagen del entorno sobre los fotorreceptores, y cada uno de ellos responde a la intensidad de luz que incide sobre él. Mediante un fenómeno de *convergencia*, muchos fotorreceptores, en la mayor parte de la retina, concentran información visual sobre un número notoriamente inferior de células ganglionares. Sólo en la fovea este número es aproximadamente igual, por lo que la visión foveal es más aguda y la visión en la periferia es mucho menos precisa. A medida que nos alejamos de la fovea, las “piezas” del mosaico serán mayores y la imagen transmitida al cerebro será más burda.

En la corteza visual primaria existe una representación retinotópica, o sea, que la estimulación de una región determinada de la retina excita las neuronas de una región

específica de la corteza visual primaria; por lo que, la estimulación de regiones adyacentes excita regiones corticales adyacentes. La superficie de la retina no se presenta de forma lineal en la corteza visual. Además, la visión foveal con máxima agudeza ocupa aproximadamente el 25% de la corteza visual.

La lesión de una pequeña porción de la corteza estriada originará un pequeño punto ciego o escotoma en el campo visual, cuya localización dependerá de la propia localización de la lesión en la corteza. Por otra parte es importante diferenciar el hecho de que la porción cortical sea la corteza primaria o las áreas secundarias o de asociación. En el primer caso, la ceguera (si la lesión es bilateral) será total, ya que no se percibirán los estímulos luminosos en su primer nivel. En el segundo caso, el individuo podrá ver objetos, letras o colores, pero no interpretará formas o significados.

En los años setenta, se efectuaron experimentos que consistían en estimular la corteza visual humana mediante electrodos. Los individuos manifestaron ver determinadas formas geométricas correspondientes al patrón de los electrodos activados en diversos momentos. Con el objeto de intentar corregir la ceguera mediante prótesis, se realizó este estudio con personas ciegas, pero la estimulación eléctrica a largo plazo causa lesiones en el tejido, por lo cual se descarta por el momento su utilización.

## II.2 La corteza cerebral

El *córtex o corteza cerebral*, representa un logro de la evolución, es uno de los capítulos con mayor éxito en la historia de los seres vivos. El grado en que un animal depende de un órgano es un índice de la importancia de ese órgano, y la dependencia de la corteza ha ido aumentando rápidamente al ir evolucionando los mamíferos (Hubel y Wiesel ,

1968).

La corteza cerebral tiene aproximadamente unos 2 mm de grosor, y es una estructura muy replegada que tiene una superficie media de unos 150000  $mm^2$  en la especie humana. Se ha calculado que posee unas  $10^{10}$  neuronas, algo más del 90% de todas las del sistema nervioso. Los cuerpos neuronales se disponen en 6 capas celulares que se numeran desde la más externa a la más interna. Alternativamente estas capas son pobres y ricas en células. A lo largo del siglo XIX y principios del XX, se cartografió bien la corteza cerebral y se vio que según la zona funcional, la estructura de las capas variaban ligeramente en cuanto al contenido celular. En la corteza, la regla es que las regiones que tienen una función más precisa o mayor discriminación ocupan relativamente más zona cortical. En la corteza visual, la región foveal de la retina tiene una representación que es 35 veces más detallada que la región periférica lejana. Sin embargo, el verdadero problema era conocer de qué manera analiza el cerebro la información, y en este sentido se acometieron dos importantes líneas de investigación que hoy en día están en pleno desarrollo:

En la primera, el primer hallazgo notable de la organización cortical fue el reconocimiento de la subdivisión en zonas con funciones muy diferentes, más o menos ordenadas cartográficamente, si bien su número ha sido motivo de gran especulación. Los anatomistas han propuesto un número amplio de regiones o áreas; Von Economo clasificó 109 áreas; Vogt clasificó 200 áreas. Mientras que el de los fisiólogos ha sido algo más modesto; fueron Campbell con solo 20 áreas; y Brodmann con 52 áreas. Así, hoy en día, se acepta la subdivisión en las 52 áreas funcionales propuestas por el fisiólogo alemán Korbinian Brodmann en 1909, la cual se puede observar en la Figura 8. Estas áreas no tienen límites exactos, y además la correlación anatómica con la función no es

tan precisa. No obstante sirve de pauta para definir localizaciones globales. La noción básica que se debe considerar es que la información sobre cualquier modalidad sensorial se transmite primero a una zona o *área cortical primaria*, y desde allí, directamente o a través del talamo, a una serie de zonas superiores o áreas de asociación.

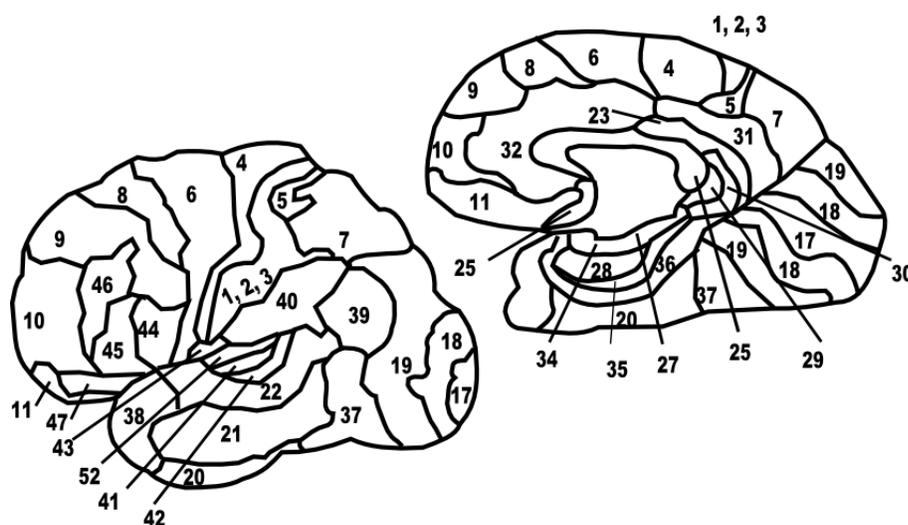


Figura 8: Cartografía de las diversas áreas corticales según Korbinian Brodmann (1909). La imagen del lado izquierdo está subdividida de acuerdo a las superficie lateral de un hemisferio cerebral y la segunda imagen, está basado en la superficie media o interna-superficial

La segunda parte del gran descubrimiento que se debe al neuroanatomista español Santiago Ramón y Cajal (Cajal , 1899), y a su discípulo Rafael Lorente, quienes pusieron de manifiesto que las operaciones que realiza la corteza sobre la información que recibe se ejecuta de forma local. En esencia, las conexiones son simples; así un conjunto de fibras nerviosas aportan información a la corteza, por lo que después de atravesar varias sinapsis, la influencia de la entrada (proyección cortical) se habrá extendido a todas las capas celulares. Finalmente, otros varios conjuntos de fibras conducen a una región concreta mensajes modificados. La naturaleza local de dichas conexiones

es común a todas las regiones corticales. La información que transporta a la corteza una sola fibra, puede en principio hacerse sentir a través de toda la profundidad, a través de unas tres o cuatro sinapsis, mientras que la expansión lateral, producida por los árboles ramificados de axones y dendritas, se limita a efectos prácticos a unos cuantos milímetros en una porción de la vasta extensión de la corteza cerebral. Así, todo cuanto realice en una determinada región de la corteza, lo hará localmente.

En la *corteza cerebral* se han identificando varias representaciones del campo visual a partir del registro electrofisiológico de los potenciales corticales evocados mediante estimulación retiniana. En relación con la tradicional clasificación de Broadmann (ver Figura 8), estas regiones o áreas son las siguientes:

- En el área 17 existe 1 región (*área visual primaria o V1*)
- En el área 18 existen 4 regiones (*V2, V3, V3a, V4*)
- En el área 19 existe 1 región (*mediotemporal V5*)
- En las áreas 20 y 21 existe 1 región (*corteza inferotemporal*)
- En el área 7 existe 1 región (*corteza parietal posterior*)

## II.3 Estructura histológica de la corteza visual primaria

### II.3.1 Estría de Gennari-Vick d'Azir en la capa IV

La corteza visual se caracteriza por una marcada estratificación orientada paralelamente a la superficie cortical, y es más delgada (1,5 cm aproximadamente) que otras áreas corticales porque, aunque en este caso sea mayor la población celular, el espacio

intercelular es más reducida. Von Economo la denominó por ello *Korniocórtex* (de *Kornio*: polvo). La principal característica que distingue la *corteza visual primaria* (área 17 o V1) es la presencia de una capa de fibras mielinizadas relativamente acelular y muy potente, que es visible sin aumento en secciones perpendiculares a ella.

Esta zona de la corteza recibe el nombre de *corteza estriada*, debido al grosor excepcional de la cuarta capa (capa IV), que es lugar donde terminan los axones de las células del cuerpo geniculado lateral. En la capa IV hay una banda de sustancia blanca que la subdivide en dos partes (IVa y IVc). esta banda intermedia (IVb) constituye la *estría de Gennari-Vick de'Azir* descrita por primera vez por Francesco Gennari en 1782. Esta capa es un plexo fibroso intracortical, mielinizado, compuesto por axones, orientados horizontalmente, y constituido por células piramidales y estrelladas (Lund , 1973). Numerosas conexiones unen el área 17 con las *áreas preestriadas* (18 y 19) o *áreas de asociación visual (áreas visuales secundarias)*.

### II.3.2 Tipos celulares en la corteza visual

La corteza visual tiene dos clases celulares fundamentales:

1. *Células piramidales*. De soma grande y con espinas dentríticas largas. Son neuronas de proyección, cuyos axones proyectan a otras regiones cerebrales la información. Además, son excitatorias, y la mayor parte de la veces secretan *glutamato y aspartato*.
2. *Células estrelladas (granulares)*. De soma pequeño, las hay aquellas que presentan espinas (*células estrelladas espinosas*). Como las piramidales, las cuales son excitatorias y secretan los mismos neurotransmisores. Otras no las presentan y se denominan *células estrelladas lisas*. Éstas son inhibitorias y la mayor parte

secretan *GABA* (*ácido gama amino butírico*).

### II.3.3 Estratificación de la corteza visual

Del mismo modo que los axones de las células ganglionares proyectan una representación espacial precisa de la retina sobre el cuerpo geniculado lateral, éste proyecta una representación similar, punto por punto, sobre la corteza visual. Como el resto del neocórtex visual se estratifica en seis capas y varias subcapas que han sido numeradas desde el exterior al interior según la Figura 9: I, II, IIIa, IIIb, IVa, IVb, IVc alfa, IVc beta, Va, Vb, VI. Así, estas capas contienen los núcleos de los cuerpos celulares y sus árboles dentrícos, que aparecen como zonas oscuras o claras, en secciones de tejido tratadas con tinciones<sup>1</sup> específicas de cuerpos celulares.

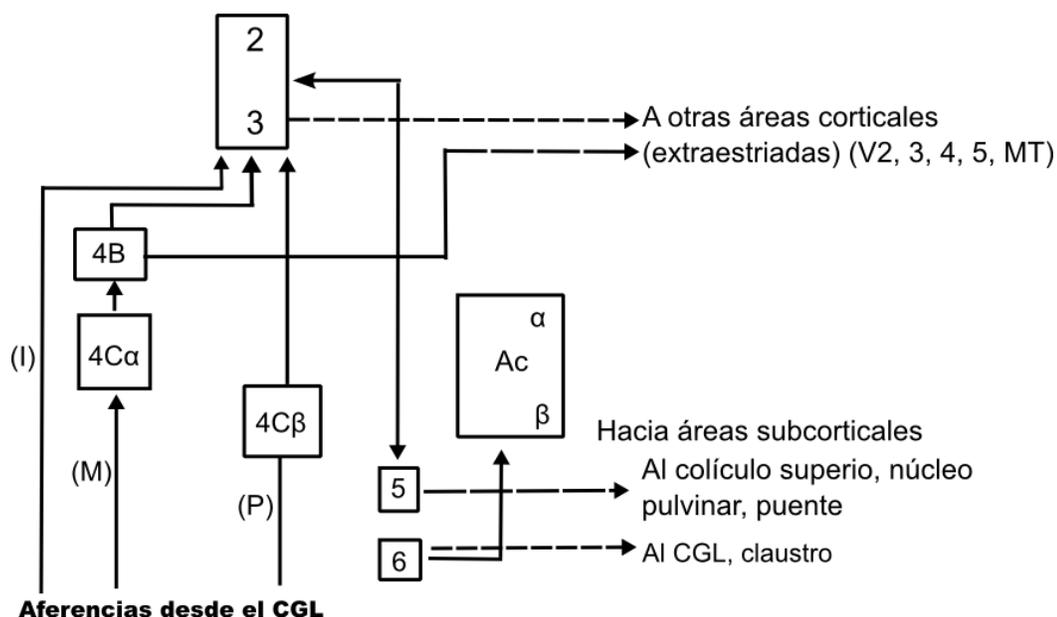


Figura 9: Estratificación de la corteza visual. Se representan las conexiones de los sistemas parvocelulares (P), magnocelular (M) y koniocelular (I) del cuerpo geniculado lateral (CGL), así como las conexiones verticales entre las propias capas del córtex y las eferencias que desde éste van a otras zonas cerebrales.

<sup>1</sup>Tinción o coloración es una técnica auxiliar en microscopía para mejorar el contraste en la imagen vista al microscopio.

### II.3.4 Circuito de retroalimentación en V1

Los axones de las neuronas del cuerpo geniculado lateral terminan sobre las células estrelladas de la capa IV. concretamente en su zona más profunda, la subcapa IVc. Los axones de la porción magnocelular (M) 1 y 2, del cuerpo geniculado terminan más superficialmente en las capas anteriores, es específicamente en la capa IVc alfa y además en la capa VI. Los axones de la porción parvocelular (P) 3,4,5 y 6, lo hacen en la capa IVc beta o en la capa IVa, que no existen en humanos (Horton y Hedley-Whyte , 1984). Además se proyectan ramificaciones menores en la capa VI.

Otro tipo de células (*Konio células o células k*), en el mono ardilla (*Saimiri sciureus*) localizadas entre las láminas del CGL (*zonas interláminas, I*), se proyectan en las capas II y III; y efectúan sinapsis con unas células agrupadas en unas estructuras denominadas blobs (*burbujas o gotas*) con respuesta al color, que serán tratadas más adelante. Sin embargo, como ha demostrado Michael (1987), para el macaco este tipo de aferencias es apenas importante, ya que masivamente las células del CGL se proyectan en la capa IVc beta, y desde allí hacen a las burbujas o gotas, véase la Figura 9.

A su vez, las células estrelladas de la subcapa IVc se proyectan principalmente a capas más superficiales de la corteza, particularmente a la capa IVb (axones de las magnocelulas), a las capas I (parvocelulares), II y III en ambos sistemas y a la capa VI (Hubel y Wiesel , 1972). Las células en la capas II y III proyectan a las piramidales de la capa V, las cuales envían axones colaterales a células piramidales de la capa VI. Éstas completan el circuito local excitatorio enviando axones colaterales a la capa IV para excitar a las células estrelladas inhibitorias (lisas). Asu vez estas células contactan y modulan las respuestas de las células estrelladas excitatorias (espinosas), con lo

que completan un circuito de retroalimentación inhibitorio. Por lo tanto, las células estrelladas espinosas distribuyen el impulso desde el CGL hacia el córtex y las células piramidales envían axones colaterales hacia arriba y hacia abajo a fin de integrar la actividad entre las capas de V1. Por otra parte, existe un gran influjo desde la capa VI al cuerpo geniculado.

## **II.4 Campos receptores en la corteza visual y detección de contornos**

### **II.4.1 Tipos neuronales en V1 según su campo receptor**

Los neurofisiólogos Hubel y Wiesel, discípulos de Kuffler, en una serie de trabajos ya clásicos efectuaron en los años cincuenta y sesenta, los primeros estudios sobre las respuestas de las células corticales visuales aisladas del gato y esta contribución les valió el Premio Nobel en 1981. Hubel y Wiesel demostraron que las neuronas del córtex visual no respondían simplemente a puntos de luz, sino que lo hacían selectivamente a características o rasgos específicos del entorno visual.

Una misma célula cortical puede responder a la excitación de una región retiniana relativamente extensa. Así, descubrieron varios tipos celulares, cada uno de los cuales respondía a un estímulo diferente dentro de su campo receptor. La mayoría de ellos respondían selectivamente a la longitud de onda, y muchos de estos tipos celulares respondían a estímulos de los dos ojos. Estos aspectos cromáticos y binoculares serán tratados posteriormente. Por lo pronto proporcionamos las siguientes consideraciones:

1. Células de campo receptor concéntrico (*estrelladas*). Las neuronas de la capa IVc de la corteza estriada tiene propiedades de respuesta similar a las del cuerpo

geniculado lateral dorsal, aunque de tamaño ligeramente superior (es decir, campos receptores del tipo centro-periferia antagónicos, con organización circular concéntrica).

2. Células con respuesta cromática (*concéntricas*). Existen fuera de la capa IV unas neuronas localizadas en las estructuras denominadas burbujas o gotas que tienen un campo circular concéntrico y que responden a estímulos cromáticos.

Las neuronas de otras capas tienen propiedades mucho más interesantes para la detección de formas y contornos. Los nombres actualmente aceptados para los principales tipos de neuronas en el córtex visual provienen en realidad del tipo de su campo receptor. Así, al hablar de células simples o células complejas, nos referimos a *células con campo receptor simple o campo receptor complejo*. Estos campos receptores, tienen generalmente una forma alargada.

En principio, estos tipos neuronales estarían jerarquizados según una complejidad ascendente, basada asimismo en una organización anatómica y de situación. Si bien actualmente los neurofisiólogos se replantean esta jerarquía, como base de las características de la organización del córtex visual, pueden aceptarse los criterios de clasificación de Hubel y Wiesel como sigue:

1. *Células simples (Células S)* (Hubel y Wiesel , 1962). Descubiertas en la corteza visual del gato. Responden con mayor intensidad a una línea recta o una franja luminosa frente a los ojos del animal, con una *orientación* determinada. Si se cambia la orientación de la línea, la respuesta de las células se hace menos intensa, véase Figura 10. La selectividad de las diferentes neuronas varía pero en general

su respuesta decae cuando la línea se inclina más de  $10^\circ$ . Las células simples están localizadas en su mayor parte en la capa IVb y algo menos en la capa VI.

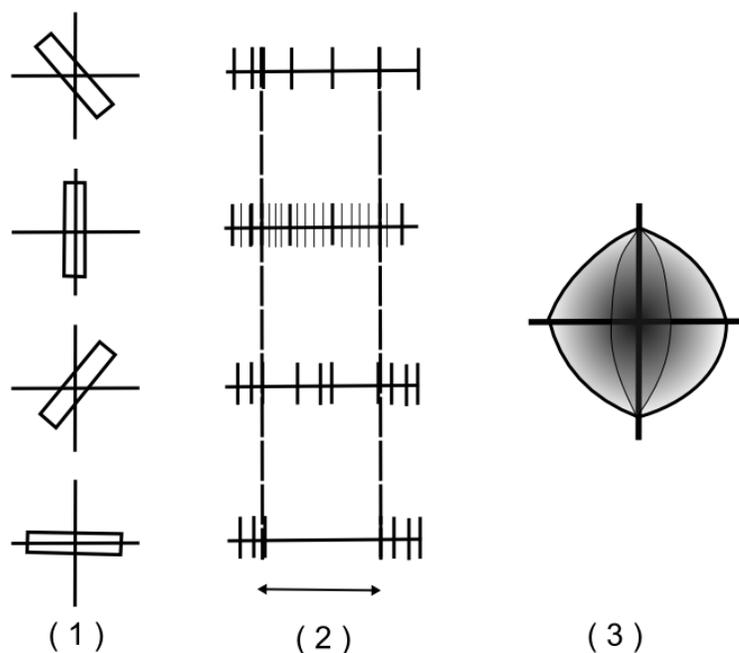


Figura 10: Respuestas de una célula simple a una barra con diversas orientaciones (de Hubel y Wiesel , 1959).

Los campos receptores de estas células tienen regiones céntricas y regiones excéntricas (centro y periferia), del mismo modo que las del cuerpo geniculado y las ganglionares, pero el campo se encontró más grande y alargado, en otras palabras no circular. Así, una barra luminosa excita a la célula si se sitúa en el centro del campo receptor, pero la inhibe si se desplaza a la periferia del mismo. La distribución de flancos excitadores-inhibidores en los distintos campos receptores simples puede no ser simétrica (Figura 11), células con campos receptivos con respuesta excitatoria (*ON*) o inhibitoria (*OFF*). Hubel y Wiesel distinguieron los siguientes tipos de células simples:

- células con centro-*ON* alargado, con gran región *OFF* a un lado y pequeño

al otro lado.

- células con una región *ON* y una región *OFF* similares a una al lado de la otra.
- células con un estrecho centro-*OFF*, con lados *ON* anchos.
- células con un amplio centro-*ON*, con estrechos lados *OFF*.

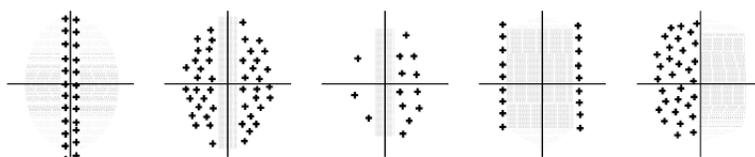


Figura 11: Tipos de campo receptor simple (de Hubel y Wiesel , 1959).

2. *Células complejas (Células C)*. Descubiertas en principio en el área 18 del gato y posteriormente en las áreas 18 y 19 del mono (Hubel y Wiesel , 1965; Hubel y Wiesel , 1968). Tienen campos receptores más grandes y el estímulo al que responden con mayor intensidad se refiere a una línea en una *orientación específica* y en *movimiento* (ver Figura 12). Estas células lo hacen de forma óptima tanto a líneas blancas sobre fondo negro, como a líneas oscuras sobre fondo blanco. También responden a los límites entre luminosidad y oscuridad. Los campos receptores complejos, a diferencia de los simples, no pueden dividirse en regiones antagonistas de inhibición y excitación.

Estas neuronas continúan respondiendo mientras la línea se desliza dentro del campo receptor, o sea, que no discriminan demasiado el lugar donde aparecía

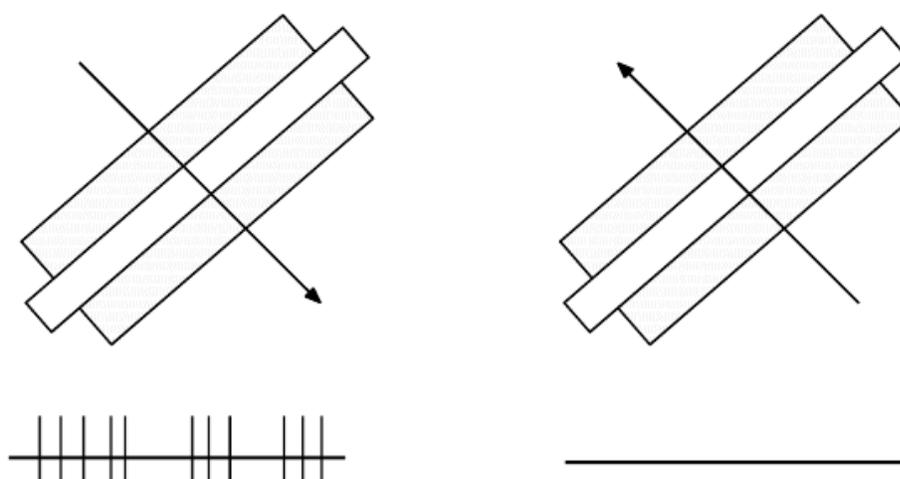


Figura 12: Células complejas con respuesta selectiva a la dirección del movimiento (de Hubel y Wiesel , 1962).

la línea en el campo visual, a diferencia de las simples. Hubel y Wiesel las describen como *células de orientación sin referencia precisa a la posición*. De hecho, muchas células complejas responden mejor al movimiento de la línea perpendicular a su ángulo de orientación. Por otra parte, las células complejas son las primeras células binoculares, ya que reciben estímulos desde ambos ojos. De acuerdo a las investigaciones de Hubel y Wiesel, las propiedades de sus campo receptor se explican por la convergencia de varias células simples con el mismo eje de orientación y con posiciones ligeramente desplazadas a lo largo de una línea horizontal en la retina. Las células complejas se localizan principalmente en las capas II y III, aunque también en las capas V y VI, además a lo largo de toda la corteza visual.

## II.4.2 Tipos neuronales según su campo receptor en la corteza circunstriada

Algunas células corticales presentaban propiedades especiales en su campo receptor, lo que llevó a Hubel y Wiesel a definir *campos hipercomplejos*, términos que luego abandonaron. Estas células con campos hipercomplejos fueron localizadas en el primate en las áreas 18 y 19 (V2, V3, V3a, V4 y V5). Si bien existen algunas en el área 17 (V1). En una célula compleja normal, la respuesta máxima se produce cuando la barra iguala la longitud total del campo receptor de la célula. Cuando el estímulo se extiende más allá del campo receptor, la respuesta no aumenta (Figura 13a).

Hubel y Wiesel hallaron células complejas con comportamiento diferente, que clasificaron en dos tipos:

1. *Células complejas “con inhibición terminal (“end stopped” cell)”* (Hubel , 1982), en principio bautizadas como células hipercomplejas de orden inferior (Hubel y Wiesel , 1968). La mejor respuesta de estas células se logró con un estímulo en forma de barra, el cual no solamente tenía una orientación específica y movimiento, sino que se requería también alguna discontinuidad, como el final de la línea, un ángulo o un vértice. Estas neuronas presentan una respuesta máxima, cuando la línea o el borde que atraviesa el campo receptor en la retina se “detiene” en uno o ambos extremos. Es decir, cuando no se extiende más allá del campo excitador en una dirección determinada (Figura 13b). Si el estímulo sobrepasa el límite, entonces la respuesta decae. Para comprender los campos receptores para este tipo celular, podemos suponer que el campo tendría un componente excitador en un lado y un componente inhibitorio al otro lado.
2. *Células complejas “con inhibición terminal total” (“completely end stopped”)*(Hubel

, 1982) o *células hipercomplejas de orden superior* (Hubel y Wiesel , 1968). Una variedad celular cortical muy interesante es aquella cuyo estímulo óptimo es una *franja en movimiento que no debe extenderse a las franjas inhibitorias del campo receptor*.

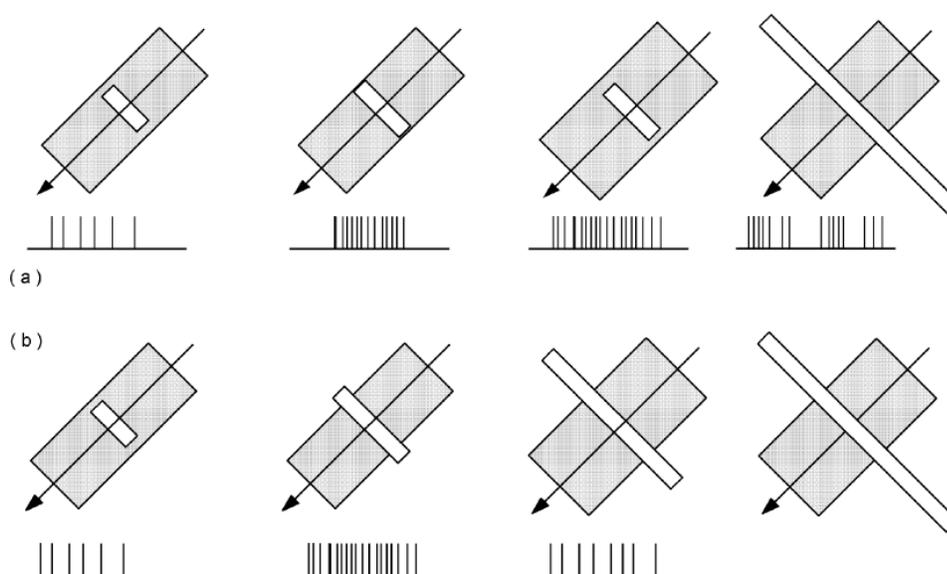


Figura 13: a) Respuestas de una célula compleja a diversa longitudes de un estímulo en forma de barra. La respuesta se va incrementando mientras la longitud de la barra aumenta aproximadamente  $2^\circ$  de arco. Más allá la respuesta no varía. b) Respuesta de una célula compleja “con inhibición terminal”. Una vez que se supera en  $2^\circ$  la ampliación de la longitud de la línea, la respuesta decae (adaptado de Hubel , 1982) .

Su campo receptor vendría determinado por una aferencia de células compleja en el centro y por dos inhibitorias a los lados. La respuesta de esta células complejas desaparece completamente si la franja se extiende fuera de la zona central (Figura 14). No nos dice con exactitud dónde está la línea en cuanto al plano de movimiento (arriba o abajo), pero sí señala que la línea no sobrepasa la posición de la región inhibidora del campo receptor.

El abandono del término “hipercomplejas” se debe a que algunos años después de

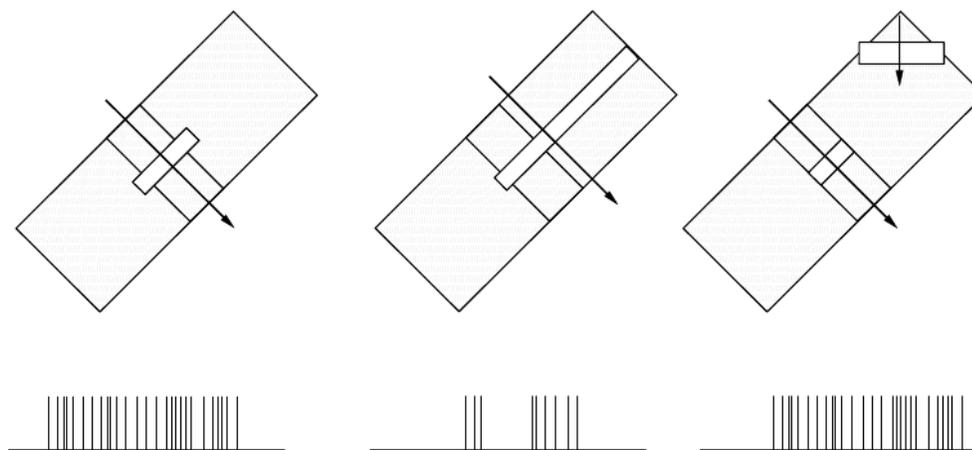


Figura 14: Cuando se estimula la región activadora central de su campo receptor, esta célula compleja con inhibición terminal total responde vigorosamente. Si el estímulo es más alargado e incide ampliamente en una de las regiones inhibitoras, la respuesta decae ostensiblemente. Cuando se cambia la orientación del estímulo y se hace incidir en una de las regiones inhibitoras, apenas existe inhibición, lo que demuestra que la orientación óptima para el efecto inhibitorio es la misma que para la región central (adaptado de Hubel , 1982).

que Hubel y Wiesel crearan el término pensando en una confluencia de células complejas hacia una jerarquía superior, después se descubrieron células simples con inhibición terminal en la corteza estriada del gato (Dreher , 1972). La inhibición terminal de estas células puede ser generada por aferencias inhibitoras del CGL o por aferencias inhibitoras de otras células simples que flanquean la región activadora de las células complejas “con inhibición terminal”.

Las neuronas corticales aparecen pues como verdaderos *detectores de contornos* según la hipótesis que Barlow propuso en el año 1972, que fue una extensión de la propuesta de Hubel y Wiesel. Los términos “simple” y “complejo”, sugieren una

jerarquización en la detección de la forma (Hubel y Wiesel , 1962), de manera que las células simples proporcionan el influjo mediante convergencia a las células complejas, y éstas, a su vez, a las complejas “con inhibición terminal”. Las células complejas “con inhibición terminal” reflejan una mayor especificidad, con una nueva introducción de la *inhibición lateral*. Indican la *importancia de la posición dentro del campo receptor*.

### II.4.3 Hipótesis propuestas sobre las conexiones entre las células de la vía visual

Hubel y Wiesel ha intentado explicar neuroanatómicamente todos los tipos de campos receptores corticales, sobre la base de sinapsis excitadoras e inhibitoras. Si bien esos modelos son ya antiguos, no existen hasta la fecha aportaciones novedosas que clarifiquen más las funciones de las células simples y complejas, por lo que se describirán tal y como fueron expuestos en los años setenta.

*Campos receptores simples (células simples)*. Estos autores sugieren que el campo en forma de banda de las células corticales simples podrían explicarse si varias células estrelladas, las cuales reciben entradas casi 1:1 respecto de las neuronas del cuerpo geniculado lateral con campos receptores concéntricos superpuestos, estuviesen conectadas de forma que se proyectaran en una única neurona cortical simple (Figura 15). Si todas las sinapsis fueran excitadoras (encendidos), la neurona cortical simple sería activada de forma óptima por una franja luminosa en un ancho aproximadamente igual a los diámetros de los *centros encendidos* de los campo receptores ganglionares. La franja luminosa estaría rodeada por una penumbra inhibitora. Este tipo de campo coincide con el de una *célula simple*.

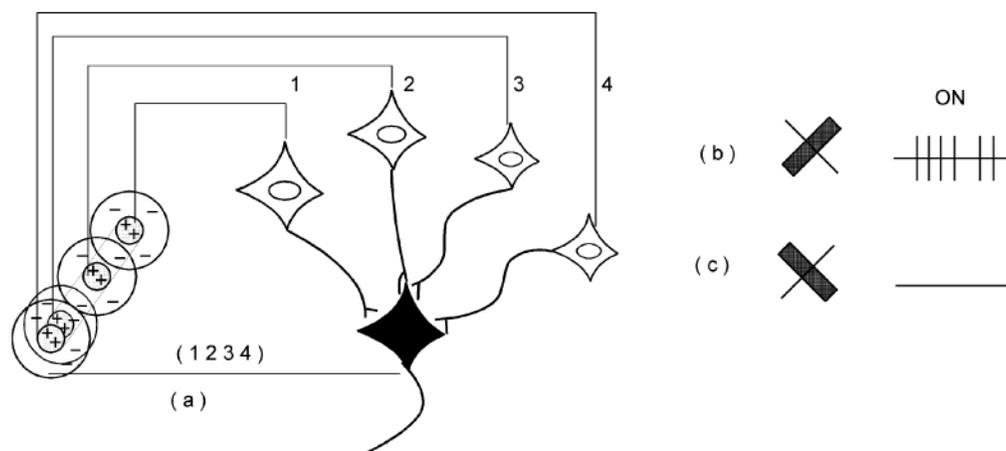


Figura 15: Explicación de cómo a partir de varios campos receptores de células estrelladas, puede formarse un campo receptor alargado correspondiente a una célula simple. a) Convergencia de cuatro neuronas estrelladas (capa IV) en una simple. b) Estímulo en barra con orientación óptima, que da la máxima respuesta. c) Estímulo con orientación ortogonal al anterior, que da respuesta nula (adaptado de Hubel y Wiesel , 1962).

*Campos receptores complejos (Células complejas).* Hubel y Wiesel los explican mediante la combinación de campos receptores simples, ver Figura 16a. Las señales procedentes de un cierto número de células con campos simples convergen sobre una célula cortical de orden superior. Cada una de ellas tiene un campo receptor en forma de banda, con la misma orientación, pero localizado en partes de la retina ligeramente separadas. La selectividad en la orientación se explicaría admitiendo la existencia de interneuronas inhibitorias (Barlow y Levick , 1965) (ver Figura 16b). Si los campos receptores simples están situados lo suficientemente cerca los unos de los otros, se entiende cómo un borde correctamente orientado, el cual incide en cualquier lugar del campo, y activa la célula cortical compleja.

*Campos complejos con "inhibición terminal" (parcial o completamente)* , Hubel y Wiesel sugieren que las propiedades de un campo receptor de este tipo podrían

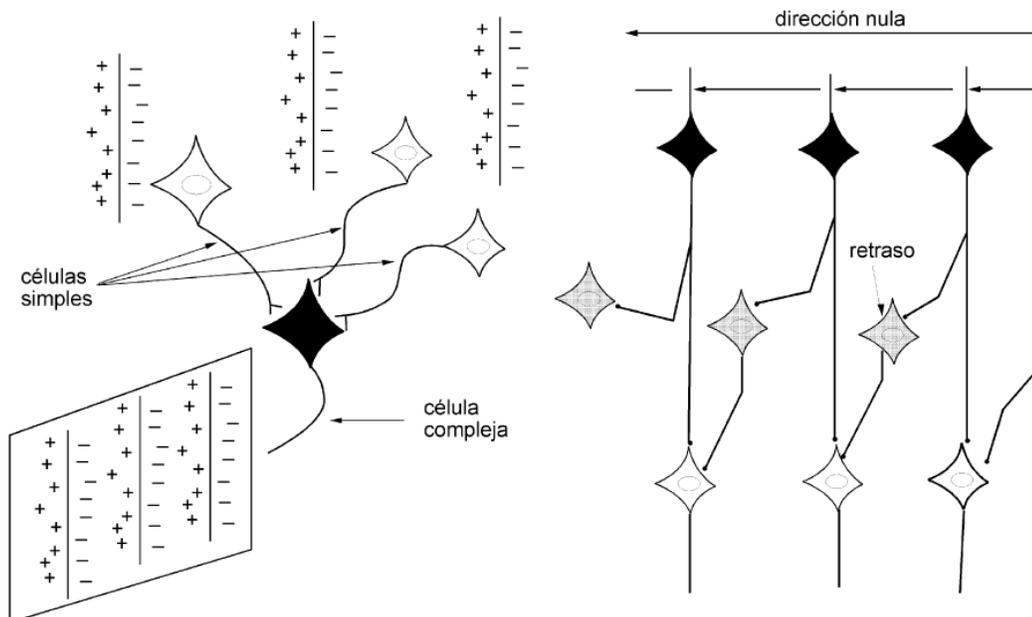


Figura 16: a) Convergencia de tres células corticales simples en una compleja. El campo receptor de ésta no discriminará exactamente una posición dentro de su campo receptor relativamente extenso. Si el estímulo sale de él habrá respuesta inhibitoria (adaptado de Hubel y Wiesel , 1962). b) Por otro lado, al ir excitando sucesivamente células simples, detectará movimiento, de forma diferente si lo hace de izquierda a derecha que de derecha a izquierda, ya que activará en un orden diferente las células simples (Barlow y Levick , 1965).

explicarse si las señales de dos células complejas normales fuesen enviadas a una célula compleja con cese en el borde según dos posibilidades:

1. Tal como se ve en la Figura 17a, se supone que uno de los campos de células complejas está conectado a la célula compleja “con inhibición terminal” a través de una sinapsis excitadora (campo a) y el otro que incluye al anterior, por una sinapsis inhibitoria (campo b). Cuando se estimula la zona “a”, la respuesta inhibitoria es mínima, mientras que si el estímulo es más alargado, la respuesta de la célula de campo receptor “b”, será muy vigorosa.
2. Otra posibilidad queda reflejada en la Figuar 17b: la célula compleja ”con inhibición terminal” recibiría entradas de tres neuronas complejas, dos inhibitorias

y una excitadora. Un estímulo orientado que provoque una excitación en la zona central, evocará una inhibición máxima en las zonas circundantes.

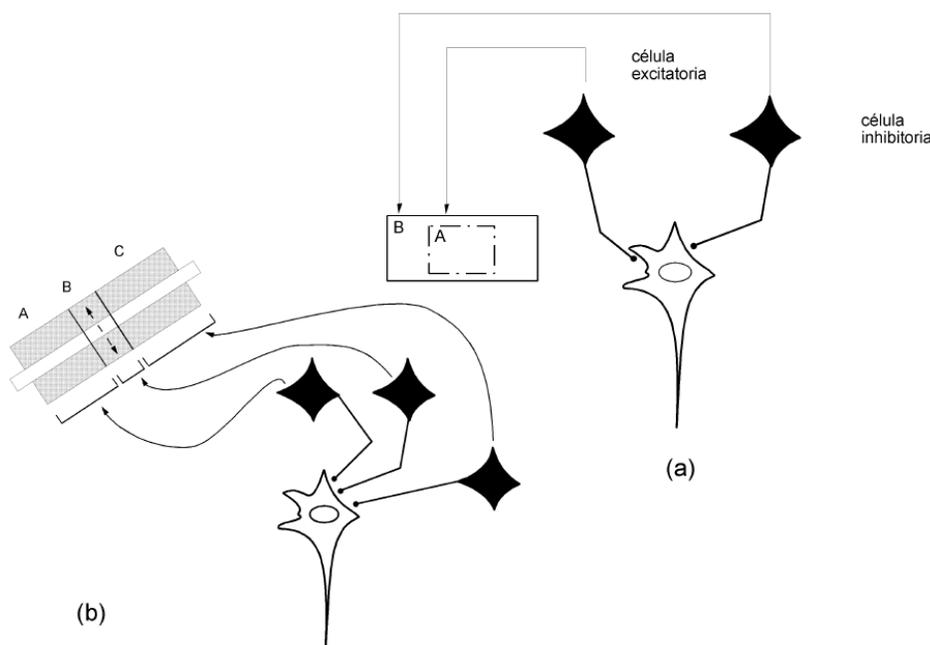


Figura 17: Modelos alternativos para explicar el campo receptor de una célula compleja con “inhibición terminal”. a) La célula inhibidora “B”, cubre enteramente el campo receptor. La célula inhibidora apenas responde cuando se estimula la región a mediante una línea corta, pero lo hace muy vigorosamente, si se sobrepasa esta región en uno o ambos extremos mediante una línea más larga. b) Convergencia de tres células complejas ordinarias en una de “inhibición terminal”. La célula que determina la región central del campo receptor es excitatorio. (A), mientras que las dos que determinan las regiones adyacentes, son inhibitorias (B y C) (adaptada de Hubel , 1982).

Pueden considerarse dos tipos de estímulos óptimos para una célula (simple o compleja) con “inhibición terminal”, según:

1. El estímulo óptimo para una célula en una zona activadora adyacente a una inhibidora, se define de acuerdo a un ángulo determinado por un borde<sup>2</sup> o una

<sup>2</sup>Modelo HMAX (Riesenhuber y Poggio , 1999)

esquina<sup>3</sup> (Figura 18a).

- Un estímulo óptimo para una célula con “inhibición terminal”, que tenga una zona excitadora central y dos inhibitoras a los lados sería una línea corta con una orientación y dirección de movimiento determinados o bien una línea que se curve de forma adecuada en la zona central y no en las adyacentes (más de 20°ó 30°) como se muestra en la Figura 18b.

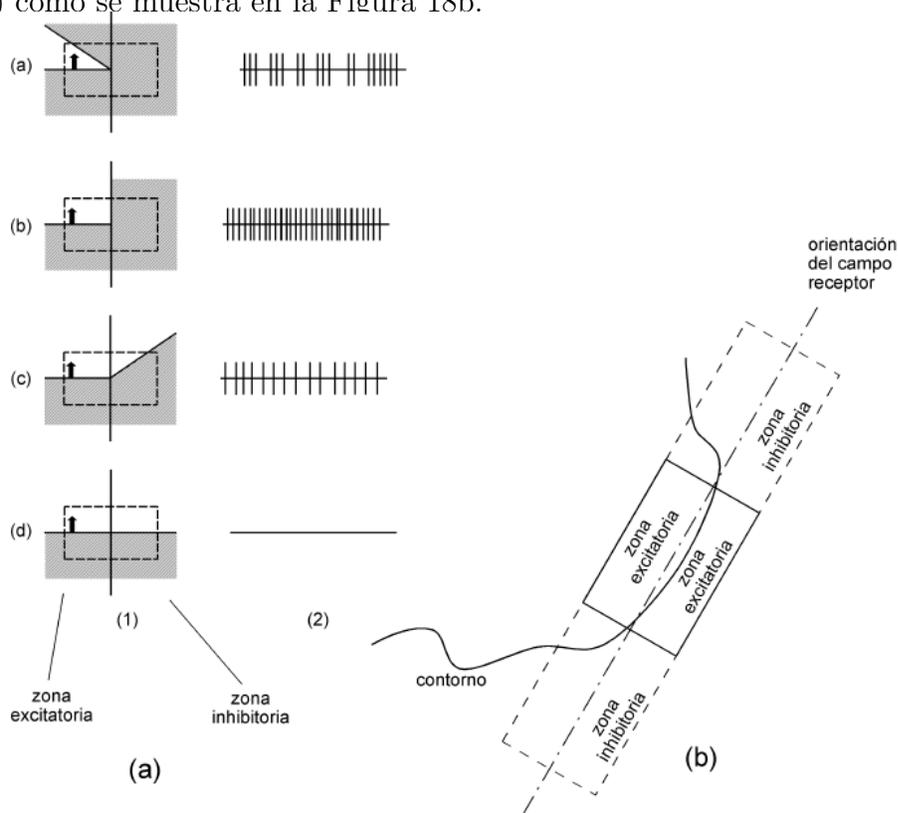


Figura 18: a) Es estímulo óptimo para esta célula es un ángulo en movimiento y orientado (90°), que no invada la zona inhibitoria de la derecha (segundo registro) (adaptado de HuWe:1965). b) Línea curva: estímulo óptimo para una célula que tenga su campo receptor conformado por una zona excitatoria flanqueada por dos inhibitorias (adaptado de Hubel , 1982).

<sup>3</sup>Nota: este aspecto justifica nuestro enfoque centrado en la utilización de detectores de puntos de interés (Trujillo y Olague , 2006)

En efecto, una línea curva será analizada por las células complejas como descompuesta a partir de varias líneas rectas con diversas orientaciones a las que responderían células simples específicas. Puede, pues, concluirse que las “células con inhibición terminal” son detectoras de curvas, esquinas, o de súbitos límites de líneas. Como propuso Attneave en 1954 (Attneave , 1954), la información visual se concentra en los contornos y más particularmente allí donde el contorno cambia de dirección. Un ejemplo es el dibujo de un gato durmiendo, realizado a partir de la abstracción de 38 puntos en los que se da la máxima curvatura y luego conectados mediante líneas rectas (ver Figura 19).

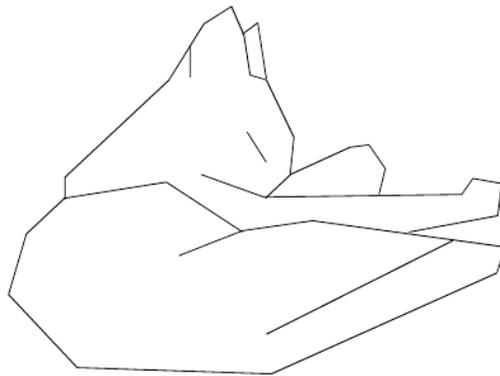


Figura 19: Dibujo de un gato durmiendo realizado abstrayendo 38 puntos de máxima curvatura y uniéndolos mediante líneas rectas (adaptado de Attneave , 1954).

## II.5 Organización modular de la corteza visual, Percepción de la forma y movimiento

### II.5.1 Organización modular (columnar) en la corteza visual primaria (V1)

Actualmente la mayor parte de los neurofisiólogos que investigan el cerebro, lo conciben organizado en módulos o unidades funcionales (Hubel *et al.* , 1976), también denominadas hipercolumnas. Estos módulos varían en tamaño según el área cortical, desde cientos de miles a algunos millones de neuronas. Concretamente, la corteza visual primaria consiste en unos 2500 módulos, cada uno de los cuales tiene aproximadamente 0.5 x 0.7 mm y contiene unas 150000 neuronas (Livinstone y Hubel , 1988).

Cada módulo o hipercolumna procesa una porción concreta del campo visual, las cuales constituye las piezas del “mosaico” de la corteza visual primaria. La organización en columnas de la corteza cerebral ya había sido sugerida hace tiempo mediante los estudios morfológicos en la corteza somatosensorial. Hubel y Wiesel confirmaron esta organización en la corteza visual. Aunque la corteza estriada presenta una citoarquitectura homogéna, mediante técnicas bioquímicas y electrofisiológicas combinadas, han sido demostradas su organización en tres tipos de columnas funcionalmente diferentes.

### II.5.2 Columnas de orientación

Si se introduce perpendicularmente a la corteza visual un microelectrodo, que penetra a través de varias capas, la orientación preferencial del estímulo al que responden las neuronas es la misma (ver Figura 20). Pero las células situadas lateralmente a pocos milímetros del electrodo responden a un estímulo con orientación diferente. Así, si una

neurona en un lugar determinado responde mejor a una línea orientada en un ángulo de  $40^\circ$ , otra neurona muy próxima a aquella responderá mejor a una línea con una orientación de  $50^\circ$ . Es decir, que las células de la corteza visual están dispuestas en una serie de columnas verticales relacionadas con la orientación del estímulo. Se les ha denominado *columnas de orientación*, ver Figura 21. Cada 20-50 micrómetros de desplazamiento lateral del electrodo se pone de manifiesto neuronas que responden a barras o líneas rotadas  $10^\circ$ , especialmente en las capas II y III.

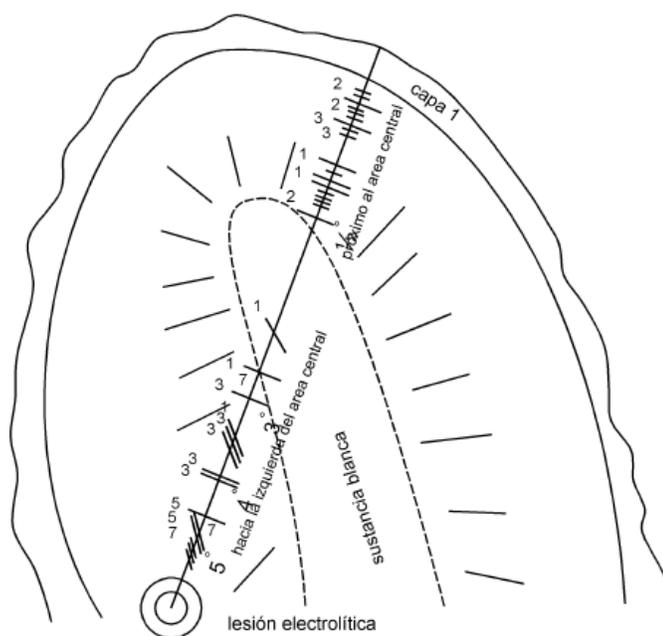


Figura 20: Ejemplo de la reconstrucción de un registro electrográfico cortical en el gato. En la porción superior de la figura, el electrodo sigue aproximadamente una columna de orientación. Una vez que atraviesa la sustancia blanca y debido al plegamiento de la corteza, atraviesa sucesivamente de manera oblicua varias columnas celulares que responden a orientaciones diferentes (Hubel y Wiesel, 1962).

Las orientaciones preferenciales de las columnas vecinas difieren de forma sistemática. Al ir cambiando de una columna a otra a través de la corteza introduciendo

el electrodo en forma oblicua, aparecen cambios secuenciales en la orientación preferencial de  $5^\circ$  a  $10^\circ$ . Puede especularse, por tanto, que para cada campo receptor de una célula ganglionar existe una colección de columnas en una pequeña zona de la corteza visual que representa la orientación preferencial posible a pequeños intervalos en los  $360^\circ$ .

Las columnas de orientación fueron identificadas y localizadas mediante un marcador bioquímico: la *2-desoxiglucosa radiactiva (2-DG)* (Wong-Riley , 1989). La captación de este derivado de la glucosa es proporcional a la actividad de las neuronas. A diferencia de la glucosa normal, la 2-DG ni puede ser metabolizada y además no abandona la célula una vez que ha entrado en ella. Empleando esta técnica en animales, principalmente primates del género *Macaco* expuestos a estimulación visual orientada de modo uniforme, i.e., líneas verticales. Finalmente, para ver la validez del experimento, el mono fue sacrificado y su cerebro cortado histológicamente. Estos cortes del cerebro muestran una organización de columnas intrincadamente curvas pero espaciadas uniformemente en una amplia zona de la corteza visual reflejada por la tinción del bioquímico.

### II.5.3 Columnas de dominancia ocular

Las células del cuerpo geniculado lateral, las de la capa IV (IVc alfa y IVc beta), y las células simples, reciben información solamente de un ojo, es decir, son estrictamente *monoculares*. Sin embargo, aproximadamente el 80% de las células complejas, correspondientes a las otras capas reciben información de ambos ojos y son, por tanto, *binoculares*. Los impulsos son idénticos o casi idénticos en lo que respecta a la porción del campo visual involucrado y su orientación preferencial. No obstante, difieren en intensidad, de tal forma que entre células cuyos impulsos provienen totalmente del ojo

homolateral o contralateral, existe toda una gamma de células influidas a diferentes intensidades por ambos ojos.

Las células influidas por un ojo se encuentran en *columnas de dominancia ocular* vertical que se alternan con columnas de células influidas por el otro ojo (Figura 21). Como existían respuestas binoculares en células complejas debe suponerse que estas columnas de orientación deben intercambiar de alguna forma información monocular. Las células corticales que tienen preferencia similar de orientación estaban ordenadas en columnas, de modo que cuando el electrodo se desplazaba en dirección ascendente hacia el cerebro, de forma perpendicular a la corteza cerebral, las células mostraban preferencia por las barras que se encontraban en la misma orientación.

Hubel, Wiesel y Le Vay en 1977, inyectando *prolina radiactiva* en un ojo, y mediante autorradiografía en cortes de la corteza visual hallaron que las proyecciones del CGL en la corteza se disponen en forma de columnar, las cuales quedan marcadas como más claras en las columnas correspondientes al ojo inyectado y no las del ojo testigo. Estas columnas tienen una anchura de entre 30 y 100 micrómetros y unos 2 mm de profundidad, donde cada columna contiene células estrelladas con campos receptores concéntricos y neuronas con campos receptores simples y complejos.

#### **II.5.4 Gotas o burbujas**

Excepto en la capa IV, las demás capas de la corteza visual contienen agrupamientos celulares que tienen casi 0.2 mm de diámetro y que, a diferencia de las células vecinas, contienen una elevada concentración de la enzima *citocromo oxidasa*, lo que indica un elevado metabolismo, debido posiblemente a que responden a cualquier orientación espacial. Los estudios pioneros de Wong-Riley en 1978, confirmados por otros autores

en 1980, pusieron de manifiesto que la tinción<sup>4</sup> de la citocromooxidasa daba lugar a un patrón punteado de columnas oscuras que se extendían a lo largo de las capas II y III y más vagamente de las capas V y VI (ver Figura 21). A estos grupos celulares se las ha denominado *burbujas*, *gotas o grumos*, del inglés *pegs o blobs*, debido a que no tienen bien definidos sus límites.

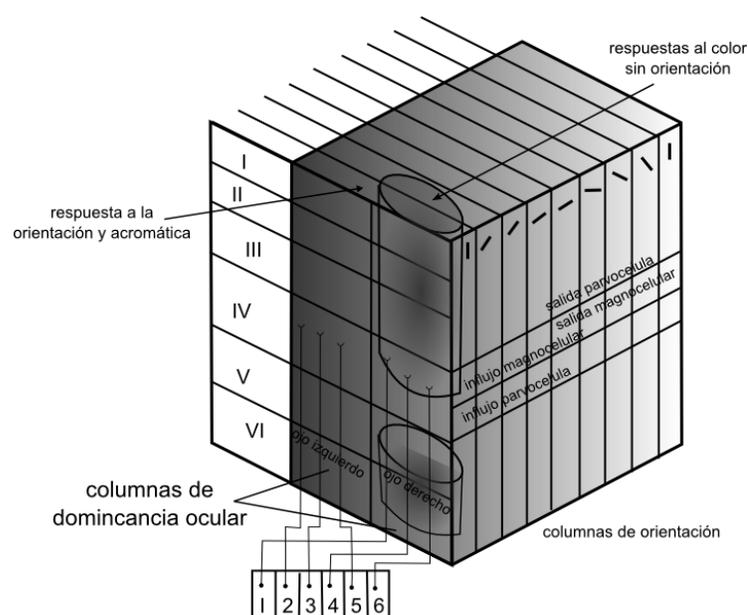


Figura 21: Esquema de una hipercolumna o módulo que incluye las “burbujas”. Se muestran asimismo las diferentes proyecciones de las células del CGL (adaptado de Livinstone y Hubel , 1984).

Las zonas entre estos bloques (aproximadamente 0.5 mm) se denominan *regiones interburbujas*. En el gato, las células del CGL contactan en la capa IV con células simples. En los primates, lo hacen en primer lugar con células estrelladas. Las células estrelladas de la capa IVc beta se proyectan básicamente a la capa III. Parece probable

<sup>4</sup>Tinción o coloración es una técnica auxiliar en microscopía para mejorar el contraste en la imagen vista al microscopio.

que las regiones de burbujas y de interburbujas reciban aferencia de subpoblaciones separadas de la capa IVc beta (continuación de parvosistema).

Las células de las burbujas se caracterizan funcionalmente por no responder a estímulos con orientación definida. Un 70% responden selectivamente a diferentes longitudes de onda, por lo que codifican el color (Livinstone y Hubel , 1976). Sus campos receptores son circulares y concéntricos. Las neuronas de las regiones interburbujas responden mayoritariamente a estímulos selectivos a la orientación y a la dirección. Si bien algunas células localizadas en los límites con las burbujas parecen responder además a estímulos cromáticos. Estos campos receptores son complejos, y el hecho de que no se hayan encontrado células simples intermedias entre la capa IVc beta y este tipo celular parece contradecir el esquema clásico de jerarquía de Hubel y Wiesel. Los módulos de Hubel y Wiesel se corresponden con al menos 12 burbujas. Las burbujas tienden a ubicarse en el centro de las columnas de dominancia ocular. Estas burbujas reciben influjo débil de las neuronas localizadas entre las capas parvo y magnocelular en el cuerpo geniculado lateral dorsal del macaco, mientras que reciben un masivo influjo desde la capa IVc beta, donde se proyectan las parvocélulas.

### II.5.5 Segregación funcional en V1

En cada columna de dominancia ocular se localizan sus correspondientes columnas de orientación y las burbujas para la discriminación cromática. *Hipercolumna, unidad funcional o módulo*, es el conjunto formado por las dos columnas de orientación correspondientes a los dos ojos, que incluyen las burbujas, y que analizan una región del espacio binocularmente (ver Figura 21). La estructura de V1 se basa en tres principios: 1) Organización retinotópica. 2) Orientación óptima del estímulo. 3) Dominancia ocular. Así se tienen funciones cuya localización es independiente en cada *unidad funcional*.

Así se tiene los siguientes pasos:

- a) Descompone el entorno visual en segmentos de línea con diversas orientaciones, lo que supone el primer análisis de forma y movimiento.
- b) Combina información de los dos ojos, que es el inicio de la visión binocular.
- c) Inicia asimismo el análisis cromático.

## II.6 Corteza visual circunscrita o de asociación (áreas visuales de asociación)

La percepción global de la escena visual no se localiza en la corteza estriada (V1). Cada módulo responde exclusivamente a una parte de la información que se presenta en el campo visual. Además, envía diferentes tipos de información a diferentes regiones de la corteza visual de asociación, cada una de las cuales contiene como mínimo un mapa del campo visual, ver Figura 22. Zeki en 1978 sugirió que este sistema permite la interacción entre tipos celulares de características similares. Así pues, para que se produzca una percepción total de la escena, la información de los módulos individuales debe ser combinada. Esta combinación se produce en la *corteza visual de asociación*. Esta corteza de asociación se extiende en parte alrededor de la corteza estriada (*corteza preestriada o circunstriada*), en una pequeña porción del lóbulo temporal (*corteza temporal inferior*), y además, a determinadas regiones de la *corteza parietal*

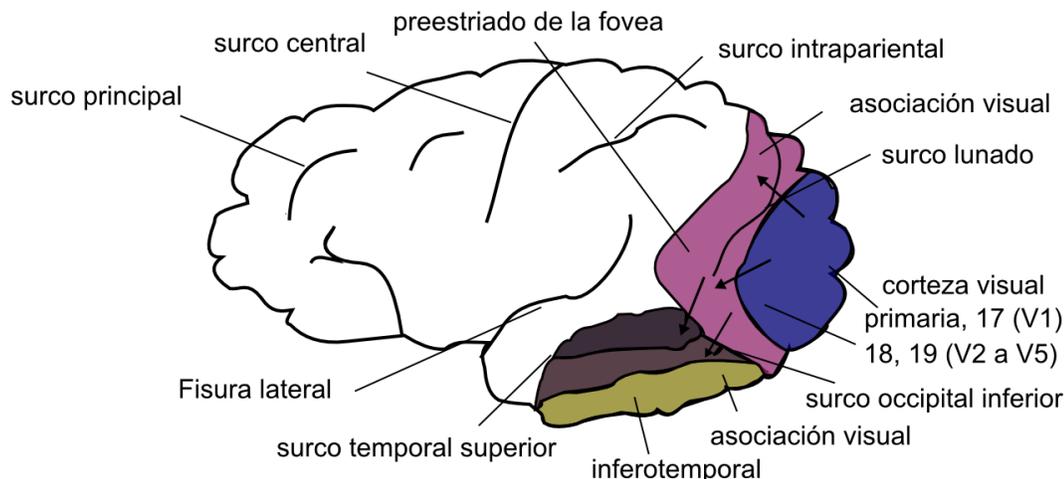


Figura 22: Relaciones de la corteza visual primaria con otras áreas (inferotemporal, circunscrita y campo ocular frontal) en el macaco.

### II.6.1 División funcional de la corteza preestriada o circunstriada

Las neuronas de la corteza estriada envían axones a otras regiones de la corteza, y en primer lugar, al primer nivel de la corteza visual de asociación. llamada la *corteza preestriada*. Aunque anatómicamente se sitúe por “delante” de la corteza estriada. el hecho de que el análisis de la información se efectúe “después” de que pase por la corteza estriada, han llevado a que algunos autores se refieran a ella como la *corteza circunstriada*. La mayor parte de las investigaciones en la corteza preestriada han sido realizadas por el equipo de Semir Zekir en el decenio de 1978 a 1988, en paralelo a las de Margaret Livingstone y David Hubel, aproximadamente en el mismo periodo. Actualmente, la corteza extriada, área 17, se denomina V1, ya que es la única zona cortical donde se proyecta el cuerpo geniculado lateral dorsal. Las neuronas de V1 envían axones a tres regiones de la corteza preestriada, las cuales son las *áreas V2*,

V3 y V5 (Zeki , 1980) (ver Figura 23). Además en otra región de la corteza preestriada, el área V3A, recibe proyecciones de las neuronas de V3, pero no directamente de V1. Otra área es la V4, la cual recibe un influjo de la V2, pero no directamente de la V1.

Cada una de estas cinco áreas; V2, V3, V3A, V4 Y V5, contiene una o más representaciones del campo visual. Según sus conexiones con las neuronas de V1, o entre ellas mismas, sus neuronas van a responder a rasgos diferentes de la escena visual (Zeki , 1989; Zeki y Shipp , 1988). Recordemos que las células ganglionares (M) o parasol proyectan a las capas magnocelulares del cuerpo geniculado lateral dorsal y las enanas proyectan a las capas parvocelulares (P). A su vez, las magnocélulas y las parvocélulas se proyectan a los sistemas

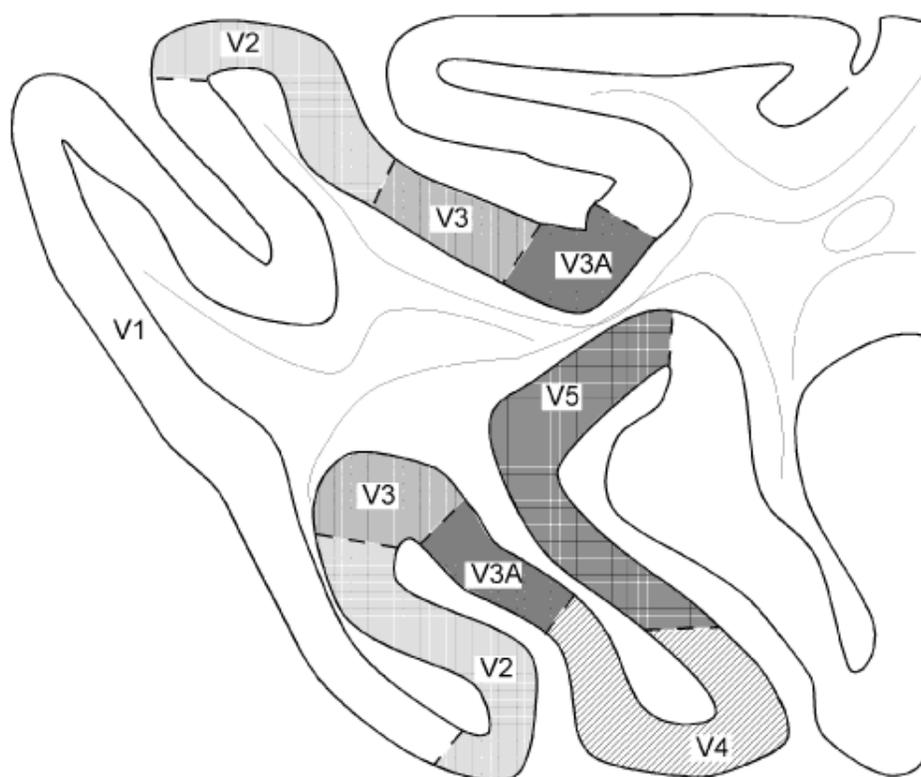


Figura 23: Localización del área visual primaria (V1) y de otras áreas del procesamiento de la información visual (V2, V3, V3A, V4 y V5) en el córtex visual del macaco (adaptado de Zeki y Shipp , 1988).

## II.6.2 Organización por columnas y segregación funcional en V2

Al igual que la tinción de la *citocromooxidasa* (Wong-Riley , 1989), había puesto de manifiesto las burbujas en la corteza estriada, empleando la misma técnica. Así, varios investigadores relevaron la presencia de bandas delgadas y gruesas en el *área V2* (Livinstone y Hubel , 1976; Tootell *et al.* , 1983). Por lo tanto, la porción de corteza circunscrita denominada V2, aparece diferenciada en dos tipos de estrías; unas *oscuras*, que según su anchura se denominan *gruesas (anchas)*, y otras *delgadas*, que están separadas por unas bandas de anchura uniforme y *claras (pálidas)*. Como pusieron de manifiesto en (Livinstone y Hubel , 1987), las características de las regiones del área V2 son muy diversas. Las bandas claras reciben proyecciones de las regiones interburbujas. A su vez, las bandas gruesas se proyectan en V3 o en V5. las bandas finas lo hacen en V4, donde asimismo se proyectan las bandas claras. Esta separación anatómica supondrá una organización funcional en la corteza, que separa en principio los diversos aspectos de la información visual.

## II.6.3 Análisis de la forma dinámica en V3

El área V3 es continuación del sistema magnocelular y es posible que realice el análisis de la forma dinámica. Además, recibe influjo de la capa IVb de la corteza estriada y de las bandas gruesas de V2. Sus neuronas son sensibles a la orientación, pero no al color (Zeki , 1989).

Hallazgos posteriores parecen demostrar que algunas células responden selectivamente al color (Essen *et al.* , 1986). Quedan por dilucidar las diferencias funcionales entre V3 y V3A.

#### II.6.4 Análisis cromático en V4

*Sistema cromático puro (color e intensidad luminosa)*. Según (Zeki , 1980), el área V4 parece estar especializada en la percepción del color. Así recibe influjo de las bandas delgadas del área V2 y muchas de sus células son selectivas para la longitud de onda.

El *Sistema de la forma asociada al color*, recibe un influjo entre las regiones asociadas a las bandas del área V2 (bandas claras) y algunas neuronas muestran sensibilidad a la orientación (Zeki y Shipp , 1988). Parece, pues, que esta región analiza asimismo la forma asociada al color.

#### II.6.5 Análisis del movimiento en V5 (MT)

El *área V5 (MT)* se localiza en la ladera posterior del *surco temporal superior* y está especializada en el análisis del movimiento. Ésta región recibe influjo únicamente del sistema magnocelular. Así, de una forma directa recibe las proyecciones de las células complejas sensibles al movimiento de la capa IVb en la corteza estriada, de las bandas gruesas del área V2 y del área V3. También del colículo superior directa e indirectamente a partir de un relevo sináptico en el *núcleo pulvinar* talámico.

Zeki y Shipp en 1988 inyectaron peroxidasa de rábano en el área V5 del macaco y observaron que al transportarse la enzima se origina un patrón de parches en el área V1, similar al producido por la tinción de citocromooxidasa. Pero estos parches no se corresponden con las burbujas, lo que indica que la organización de los módulos corticales responden a un patrón más complejo que el propuesto.

Parece que el influjo desde el colículo superior en el área V5 tiene una gran importancia. Rodman y Gross en 1989 observaron que la destrucción de la corteza estriada no eliminaba la sensibilidad al movimiento de las neuronas V5 (Rodamn *et al.* , 1989), pero la posterior destrucción del colículo superior sí lo hacía. Sus resultado demostraron que en las experiencias con monos que si bien éstos pueden detectar el movimiento, demuestran dificultades para evaluar su velocidad.

Entonces, junto con Albright (Rodamn *et al.* , 1989), trazaron un mapa de las características de las neuronas sensibles al movimiento en V5. Ellos encontraron que todas las neuronas de esta región respondían mejor a los estímulos en movimiento y no así a los estáticos, y que además la mayor parte de ellas eran insensibles al color o a la forma de los estímulos de prueba. Por lo tanto, la mayoría de las neuronas eran selectivas a la dirección. Comprobaron también que como la corteza estriada, el área V5 esta dividida en módulos, veáse la Figura 24.

Cada módulo consistiría en dos paralelepídeos contiguos. Moviéndose a lo largo del eje, se encontrarían neuronas cuya sensibilidad al movimiento variaría sistemáticamente en el sentido de las agujas de un reloj o en sentido contrario. Las neuronas de la zonas adyacentes de cada paralelepípedo tendrían sensibilidades al movimiento orientadas en direcciones opuestas. Este circuito permitiría al sistema visual extraer información sobre la “forma” en movimiento, es decir, la capacidad de percibir elementos que se mueven en una determinada dirección como si pertenecieran al mismo objeto.

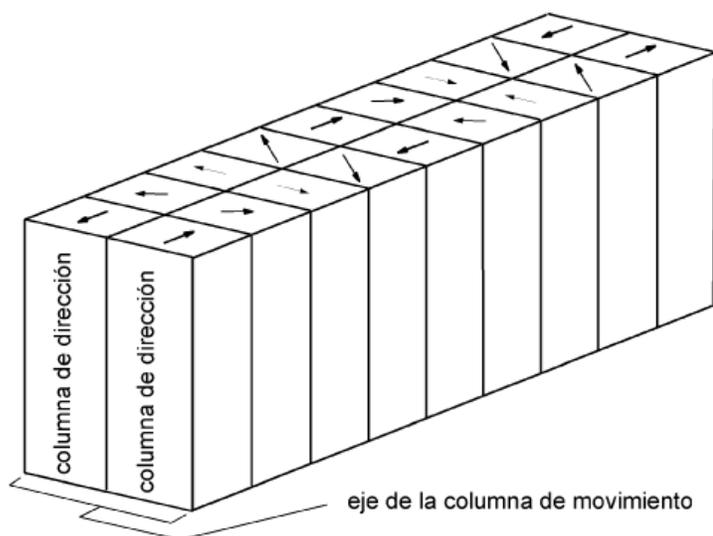


Figura 24: Módulo cortical en el área V5 (adaptado de Rodamn *et al.* , 1989).

## II.7 Corteza temporal inferior (íferotemporal)

En los primates las ejecuciones que realiza la corteza preestriada representan un grado intermedio del análisis visual. Los procesos más complejos que corresponde a los patrones visuales y a la identificación de los objetos particulares, parece tener lugar en la *corteza temporal inferior*, que se localiza en la mitad ventral del lóbulo temporal, que se denomina *circunvolución temporal inferior*. Esta área de asociación cortical recibe influjo de la corteza preestriada y de varios núcleos talámicos, especialmente del pulvinar. Es muy probable que en esta región donde converjan los análisis de forma, color, movimiento y profundidad. La corteza temporal inferior de los primates ha sido objeto de numerosos estudios que han puesto de manifiesto algunas de sus propiedades. En general, su neuronas responden mejor a los objetos tridimensionales (o a fotografías de ellos) que a los estímulos simples, tales como puntos, líneas o rejillas sinusoidales.

En Gross *et al.* , 1972 se descubrieron algunas que eran más sensibles al movimiento,

al contraste, la orientación, etc. Pero también se encontraron células que eran extremadamente particulares respecto a aquellos estímulos a los que respondían con más fuerza. Así, algunas responden mejor a fotografías de una mano, ver Figura 25, mientras que otras al perfil de la cara de un mono, y algunas otras a la visión frontal de la cara de un mono. La destrucción de la corteza visual de asociación en el lóbulo temporal de las personas causa deficiencia en la percepción visual de las formas “familiares”.

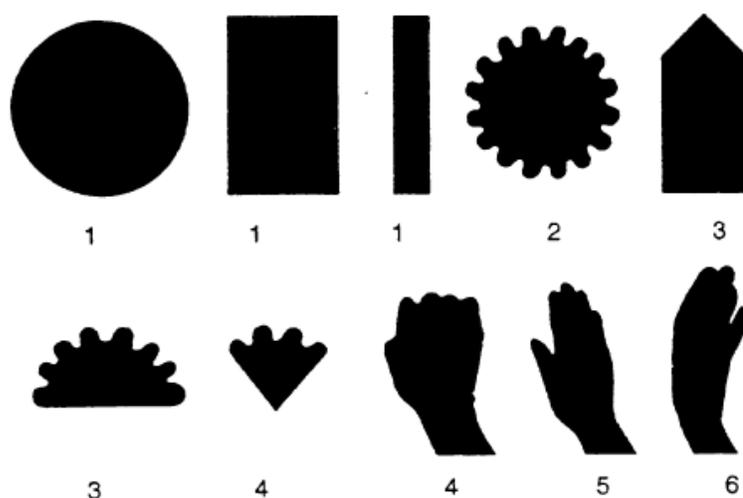


Figura 25: Experimento de Gross (Gross *et al.*, 1972). El número indica la intensidad de la respuesta en células del córtex inferotemporal de un macaco: 1 no evoca respuesta; 2-3 Ligeramente incrementado; 4-5 respuesta gradual cada vez más intensa; 5 representa el perfil de una mano humana; y 6 respuesta con máxima intensidad al perfil de una mano de macaco de su misma especie.

De esta forma, sus circuitos neuronales “aprenden” a detectar los estímulos de una forma particular, independientemente de su tamaño o localización. Iwai y Mishkin, 1969 entrenaron a macacos a discriminar entre el signo de suma y un cuadrado, reforzando sus aciertos con una breve alimentación. Extirparon en ambos hemisferios la corteza temporal, y comprobaron que los monos requerían varios cientos de ensayos para reaprender la tarea. Observaron también que pequeñas diferencias en el estímulo

impedían la ejecución de resultados correctos en los monos con lesiones en la corteza temporal inferior. Aunque los animales podían ser reentrenados a discriminar entre los estímulos originales, fallaba en su discriminación cuando se superponían a diferentes fondos.

## II.8 Corteza parietal posterior

Las áreas V3, V4 y V5 envían información además de la circunvolución temporal, a la *corteza parietal posterior*. Esta región parece estar involucrada en la percepción espacial, al recibir influjo a través de dichas conexiones. Las lesiones en el lóbulo parietal impiden la ejecución de tareas que requieren la percepción y el recuerdo de la localización de los objetos (Ungerleider y Mishkin , 1982).

## II.9 Integración final de la información visual

Zeki (Zeki y Shipp , 1988) a partir de los datos de Livingstone y Hubel (Livingstone y Hubel , 1987; Livingstone y Hubel , 1988) y de sus propias experiencias electrofisiológicas propusieron la existencia de cuatro sistemas de procesamiento de la información visual<sup>5</sup>: uno para el color, otro para el movimiento, cada uno de ellos relacionado además con un subsistema de la percepción de la forma. No obstante, este esquema conceptual que se ilustra en la Figura 26 no es del todo estricto, ya que algunas investigaciones recientes aportan datos contradictorios respecto a la especialización precisa de las áreas visuales.

- *Sistema cromático puro: Parvosistema*  $\longrightarrow$  V1 (capa IVc beta—burbujas)  $\longrightarrow$  V2

---

<sup>5</sup>Nota: esta idea fue retomada por Marr y presentada a la comunidad de visión por computadora

(*bandas finas*)  $\longrightarrow V_4 \longrightarrow$  *corteza íferotemporal*. Un sistema, masivamente interconectado desde la IVc beta y escasamente interconectado desde la interlámina del CGL (*koniocelular*), se proyecta en las propias burbujas donde las células presentan cromoselectividad; pero no responden a la orientación del estímulo. Proyectan a las bandas oscuras finas de V2 y de allí a V4, cuyas células responden selectivamente a la longitud de onda y al contraste crómico. Por lo tanto su función es la *percepción cromática pura* cuyo detalle en la cualidad cromática.

- *Sistema de la forma asociada al color. Parvosistema  $\longrightarrow V_1$  (capa IVc beta—interburbujas)  $\longrightarrow V_2$  (bandas claras)  $\longrightarrow V_4 \longrightarrow$  corteza íferotemporal.* El primer subsistema, desde la capa IVc beta se proyecta a la zona interburbujas. Éste conduce información de alta resolución sobre los límites constituidos por contraste luminosidad. Sin embargo, las neuronas de los primeros estadios de este sistema son selectivas al color, las de los niveles superiores responden a los límites generados por contraste; pero no llevan información sobre qué colores definen el límite (contraste acromático). Ni tampoco responden a direcciones particulares de movimiento. De esta forma se proyectan a las bandas claras de V2 y desde allí a V4. Dado que gran parte de la información sobre la forma de los objetos puede representarse por sus límites o bordes, puede concluirse que el parvosistema-interburbujas-bandas claras participa en la *percepción de la forma asociada al color*. Su función será el reconocimiento de letras, lectura, a fin de determinar la textura de las superficies y en definitiva descifrar “que” es el objeto y su significado.
- *Sistema de movimiento, esteropsis y localización espacial: Magnosistema  $\longrightarrow V_1$  (capa IVc alfa-capla IVb)  $\longrightarrow$  banda gruesas  $\longrightarrow V_5$  (MT)  $\longrightarrow$  corteza parietal posterior; además V1 (IVb)—corteza parietal posterior.* Las neuronas de la

capa IVc alfa se proyectan a IVb. La mayoría son células simples, con respuesta a la orientación y no selectivas para el color. Las células de la capa IVb tienen campos receptores similares a las de la IVc alfa, pero muchas de ellas son selectivas a la dirección. Las neuronas de este sistema tienen una resolución temporal muy rápida, pero sus respuestas se producen en fases o también denominadas en fásicas, y decaen inmediatamente aunque se mantenga el estímulo. Estas células no presentan cromoselectividad. Las células de la capa IVb se proyectan en las *bandas gruesas*, que muestran selectividad a la orientación. Por otra parte, en su mayor parte manifiestan una enérgica respuesta a las variaciones de *disparidad retiniana*, por lo que deben desempeñar un papel relevante en la estereopsis<sup>6</sup>. Esta vía analiza las posiciones de objetos en tres dimensiones de las coordenadas alrededor del cuerpo. Además, describe dónde se encuentra el objeto en cada instante y si está en movimiento. En el límite de la corteza parietal posterior (corteza parietooccipital), las señales se traslapan con señales procedentes de las áreas posteriores de asociación somática, que analiza la forma y los aspectos tridimensionales de las señales sensoriales somáticas.

- *Sistema de la forma dinámica. Magnosistema*  $\rightarrow V1$  (capa IVc alfa—capa IVb)  $\rightarrow V2$  (bandas gruesas)  $\rightarrow V3 \rightarrow V3a \rightarrow$  corteza íferotemporal. Además:  $V1$  (IVb)— $V3$ — $V3a$ —corteza íferotemporal. Se ocupa de la percepción de la forma de los objetos en movimiento. Es decir de “qué” son los objetos que se están moviendo. La mayoría de sus neuronas no son sensibles al color, y sí a la dirección y orientación. El hecho de que una cantidad significativa de estas neuronas responda al color, entra en contradicción con el esquema de Zeki, vea la Figura 26.

---

<sup>6</sup>estereopsis: es el proceso dentro de la percepción visual que lleva a la sensación de profundidad a partir de dos proyecciones ligeramente diferentes del mundo físico en las retinas de los ojos

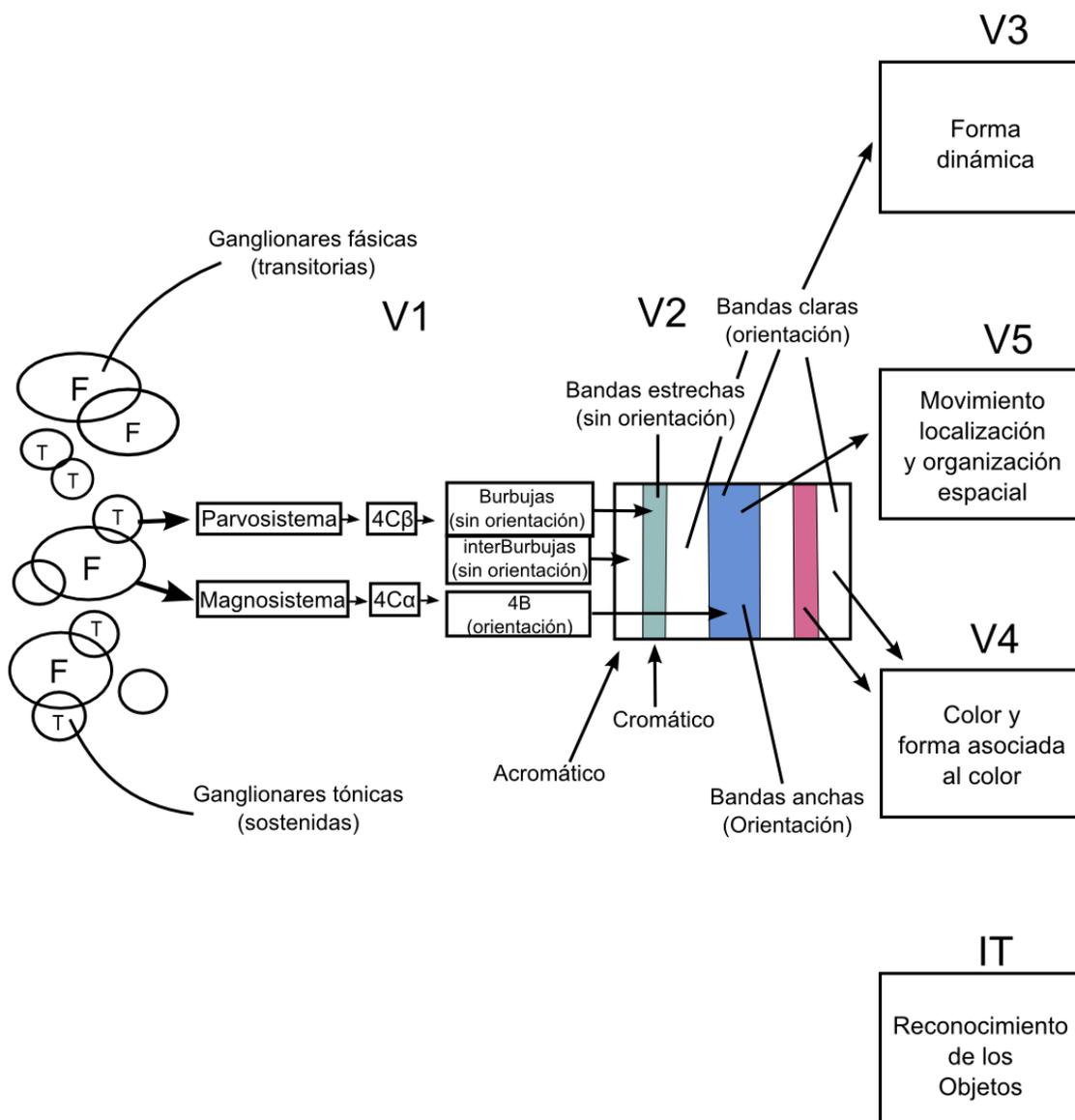


Figura 26: Vías parvo y magnocelulares desde la retina y CGL a través de V1 y V2 hasta las áreas V3, V4, V5 e IT. (adaptación a partir de los datos de Livinstone y Hubel , 1984; Zeki y Shipp , 1988)

# Capítulo III

## Programación Genética

La teoría de la evolución (Darwin , 1993) fue formulada por Charles Darwin para explicar el proceso de adaptación de las especies por medio del principio de selección natural, el cual favorece a aquellos individuos que se adaptan más fácilmente a su entorno ambiental y por ende, serán los individuos que sobrevivirán. La teoría Neo-Darwinista considera también el concepto de herencia, siendo la fuente de inspiración para los algoritmos evolutivos.

La biología ha sido desde siempre fuente de inspiración en el campo de las Ciencias de la Computación e Ingeniería para el desarrollo de métodos de optimización y búsqueda alternativas. La Computación Evolutiva (EC, por sus siglas en inglés), aplica la teoría de la evolución natural y la genética en la adaptación evolutiva de estructuras computacionales, proporcionando un medio alternativo para atacar problema complejos en diversas áreas de la ingeniería. Una población de posibles soluciones de un problema dado, es análogo a una población de organismos vivos que evolucionan en cada generación al recombinar los mejores individuos de la población y transmitir sus características de dichos individuos a sus descendientes. En este paradigma, diferentes esquemas de métodos evolutivos se han desarrollado, los cuales difieren en el tipo de estructuras que conforman la población.

Historicamente, los algoritmos genéticos fueron la primera técnica evolutiva utilizada y desde los inicios de la computación evolutiva se consideran como lo más populares. Esta técnica fue creada por John Holland (Holland , 1992) y descrita por él mismo en su libro *Adaptation in Natural and Artificial Systems*. Un algoritmo genético (Koza , 1995) transforma una población integrada por un cierto número de individuos, cada uno de los cuales tiene un cierto valor de adaptación, mediante la utilización de operaciones análogas a la genética como lo son la cruce y la mutación. Los individuos codificados en este caso se les denomina el genotipo de un individuo en biología y representan posibles soluciones al problema. El efecto indirecto que tiene normalmente el genotipo del individuo sobre su constitución (fenotipo) se reemplaza en esta técnica por el cálculo del valor de adaptación de los individuos.

Para representar el efecto que tiene el proceso de selección natural sobre una población, se emplean un gran número de distintas técnicas de selección de población (Mitchell , 1998). Sin embargo todas tienen en común el seleccionar preferentemente a los individuos con una mejor adaptación, diferenciando a un algoritmo genético de una simple búsqueda aleatoria (Spall , 2003; Banzhaf *et al.* , 1998).

El *objetivo* primordial de los *algoritmos genéticos* desarrollados por John Holland fue el estudio formal de los procesos de adaptación natural y de cómo estos mecanismos podrían ser trasladados al área del aprendizaje de máquina.

La Programación Genética (Banzhaf *et al.* , 1998) “GP<sup>1</sup>” es otra técnica evolutiva que goza de gran popularidad en la actualidad. Esta técnica fue propuesta por John Koza (Koza , 1989; Koza , 1994) a finales de los 80’s. John Koza en su libro titulado

---

<sup>1</sup>Genetic Programming, por sus siglas en inglés

*“Genetic Programming: On the Programming of Computers by Means of Natural Selection”*. Describe un método para la evolución de estructuras complejas como son los programas de computadoras. El hecho de que muchos problemas prácticos de diferentes dominios de aplicación, puedan ser formulados como un problema de determinación de un *“programas solución”*, que produzca una salida deseada cuando se tienen presentes ciertas entradas particulares, hace a la programación genética una novedosa línea de investigación. Dentro de algunas de las aplicaciones prácticas de la programación genética se tienen el modelado e identificación de sistemas, procesamiento de señales e imágenes, diseño de circuitos electrónicos, control, robótica y predicción, entre algunas otras.

### **III.1 Fundamentos de la Programación Genética**

En GP, un programa esta formado a su vez por programas, individuos, cuya estructura depende de un conjunto de funciones y terminales. Los individuos varían de una generación a otra, ya que los operadores genéticos juegan un papel fundamental en la evolución de los mismos. La representación de los datos es clave en los algoritmos genéticos, ya que manejan directamente una representación codificada de las soluciones factibles al problema. El esquema de representación modela la *“ventana”* mediante la cual se describe al problema y por lo tanto puede limitar severamente el alcance del mismo; aunque, la representación de un problema mediante cadenas de longitud fija permite resolver una amplia variedad de problemas.

Una de estas limitantes es la relación directa entre el tamaño de las cadenas utilizadas y el tamaño de la solución factible representada. Otra es la relativa rigidez de los sistemas de codificación utilizados en algoritmos genéticos. La programación

genética busca eliminar estos problemas, proporcionando un método para eliminar la relación directa entre las parte del genoma y las parte de la solución. En lugar de ello se utiliza una representación indirecta donde las cadenas pasan a ser de longitud variable e interpretarse como programas. El efecto de cada instrucción consiste en alterar la forma de la solución o cambiar el estado del sistema encargado de interpretar los individuos. Esto permite una representación mucho más compacta, donde la longitud de la cadena que representa un individuo puede acercarse a la complejidad de kolmogorov (Li y Vitányi , 1997).

Puede verse una cierta similitud entre esta codificación y el funcionamiento de la transcripción del ADN en una célula normal. Si bien la transcripción desde la secuencia de bases en el ADN a la secuencia de aminoácidos en las proteínas es relativamente directa, la complejidad e indirectación de la acción de éstas para formar un individuo es similar a la forma en que se produce la interpretación de un programa en la programación genética .

Puede observarse que la descripción del funcionamiento de la programación genética es muy similar a la de los algoritmos genéticos. La diferencia esencial radica en el proceso de evaluación: mientras los algoritmos genéticos evalúan directamente el genotipo del individuo, la programación genética ejecuta el genotipo como un programa tomando el resultado como el fenotipo a evaluar, dicho de otra manera se realiza una secuencia lógica de la lectura del programa para después evaluarlo. La frontera exacta no es clara, ya que podría tomarse a la ejecución de un programa como una forma sofisticada de evaluación (McCarthy , 1997), pero es esta la característica distintiva de la programación genética.

Los programas que pueden ser representados como grafos tiene en general estructuras complejas y de tamaño variable. Esto fuerza el desarrollo de operadores genéticos relativamente complejos ya que, en caso contrario, habría un número excesivo de individuos incapaces de ser ejecutados. Desde los comienzos del uso de las computadoras se ha pensado en automatizar el proceso mismo de programación genética, por lo que en general se extiende a expresar los problemas en un lenguaje de mayor nivel de abstracción que los existentes (Novak , 2005). Limitándose a la aplicación de operadores genéticos al desarrollo de programas, puede señalarse a (Smith , 1980) como el primer trabajo donde se realizó un desarrollo evolutivo de programas, en lugar de solo parámetros de los mismos. Sin embargo, el origen de un interés más amplio en la programación genética estuvo dado por la publicación por parte de John Koza de varios trabajos y un libro describiendo este paradigma (Koza , 1989; Koza , 1994; Koza , 1992).

### III.1.1 Conceptos Básicos

La mayoría de los sistemas de programación genética tiene en común las siguientes características:

- *Decisión Estocástica.* La programación genética usa números pseudoaleatorios para emular la evolución natural. Como resultado, la programación genética utiliza procesos estocásticos y decisiones probabilísticas.
- *Estructura del Programa.* La programación genética integra dos tipos de conjuntos: funciones y terminales. Las funciones ejecutan operaciones desde sus entradas las cuales pueden ser terminales o salidas de otras funciones. Desde el inicio de la ejecución del algoritmo, la población es inicializada con programas que son formados a partir de funciones y terminales.

- *Operadores Genéticos.* La programación genética transforma los programas de la población inicial usando operadores genéticos. La cruce entre dos individuos es uno de los principales operadores genéticos en la programación genética. Otros operadores genéticos importantes es la mutación y la cruce.
- *Evolución de la Población.* La selección basada en la aptitud, determina qué programas de la población se eligen para fungir como los padres en las siguientes generaciones.

## III.2 Terminales y Funciones

Los símbolos llamadas funciones y terminales, son las primitias a partir de las cuales se contruye un programa en programación genética. Estos juegan diferentes papeles. A grandes rasgos, en general podemos decir que las terminales proveen un valor al sistema, mientras que las funciones procesan dicho valor. Tanto a los símbolos de funciones como a las terminales, se les denomina nodos cuando la representación de los programas sean en forma de árboles, veáse la Figura 27.

### III.2.1 Conjunto de Terminales

El conjunto de terminales está constituido por:

1. **Variables** que son las estradas al programa: un individuo de la población puede ser  $f = x + 2y$ , en el cual las entradas de este programa son las variables  $x$  y  $y$  las cuales forma parte del conjunto de los símbolos de terminales.
2. **Constantes** que son suministradas a un programa: este tipo de terminales no cambian su valor durante la ejecución del programa.
3. **Nulos** son las funciones que no tienen ningún argumento o aridad.

En una representación basada en árboles, el conjunto de terminales recibe el nombre de conjunto de terminales o nodos terminales, ya que se constituyen las hojas o nodos terminales de las ramas de los árboles.

### III.2.2 Conjunto de Funciones

El conjunto de funciones está compuesto por sentencias, operadores y funciones disponibles para los sistemas de programación genética.

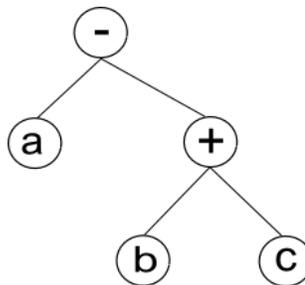


Figura 27: Rama de un árbol.

Algunos ejemplos del conjunto de funciones son las siguientes:

- *Operadores Booleanos:* AND, OR NOT, XOR.
- *Operadores Aritméticos:* SUMA, RESTA, MULTIPLICACIÓN, DIVISIÓN.
- *Funciones Transcendentes:* TRIGONOMÉTRICA Y FUNCIONES LOGARÍTMICAS.
- *Funciones de Asignación a variables:* Asignación de valores a las variables.
- *Sentencias Condicionales:* If, Then, Else; Case o Switch.
- *Sentencias de Control de Transferencia:* Go to, Call, Jump.
- *Sentencias de Ciclos:* While... Do, Repeat... Until, For... Do.

### **Selección del conjunto de Terminales y Funciones**

El conjunto de las terminales y funciones usados en la programación genética deben ser lo suficiente poderosos para ser capaces de representar la solución del problema. Por ejemplo, el conjunto de las funciones que contenga sólo el operador de adición muy probablemente no podrá resolver problemas complejos. Por otra parte, es mejor no tener un conjunto de funciones muy grande, ya que esto podría aumentar el espacio de búsqueda y por consiguiente, hacer que la solución sea más difícil de encontrar.

Una propiedad importante de los conjuntos de las funciones. Es que tengan la propiedad de cerradura. Esto es que cada función sea capaz de manejar todos los valores que reciben las entradas y produzca valores del mismo tipo. Todas las funciones deben ser capaces de aceptar todas las posibles entradas ya que de no ser así el sistema colapsaría, es decir que si los datos no fueron definidos correctamente el sistema no entenderá que hacer.

### **III.2.3 Estructura de un programa a través de programación genética**

Las primitivas de la programación genética, funciones y terminales, no son programas sino más bien componentes de los mismos. Estas deben ser integradas a una estructura antes de que éstos puedan ser ejecutados como programas. La selección de la estructura de un programa en programación genética afecta el orden de ejecución, uso de la memoria y la aplicación de los operadores genéticos al programa. En esta sección se explican las estructuras de árbol, lineales y grafos. Estas estructuras son las principales utilizadas en GP.

A continuación se muestran las representaciones más utilizadas para los genotipos

en la programación genética, las tres principales estructuras utilizadas son: estructuras de árboles, lineales y grafos.

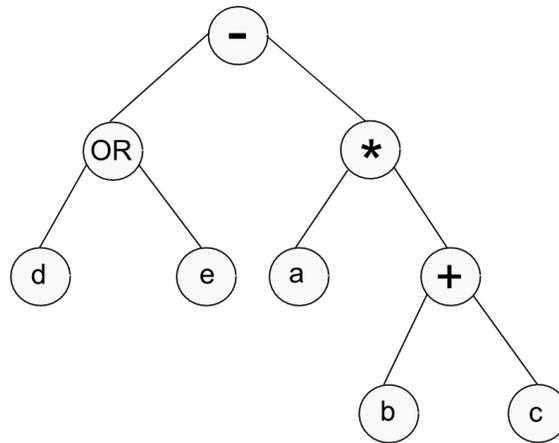


Figura 28: Estructura de un árbol.

### Ejecución de la estructura de un árbol

Para poder almacenar un programa es necesario elegir una forma de representarlo. La forma más comúnmente usada (Brameier , 2004) es la representación en forma de árbol. Esta se encuentra inspirada por el lenguaje de programación Lisp y que se emplea por John Koza en algunos trabajos pioneros del área (Koza , 1989; Koza , 1992). En esta representación, un programa se almacena como un árbol en el que los nodos terminales representan valores o funciones sin argumentos, mientras que los nodos internos representan funciones que toman como argumento los valores de sus nodos hijos. De esta forma, la ejecución se realiza en forma recursiva desde el nodo raíz, recibiendo cada nodo los valores producto de la ejecución de los subárboles descendientes, procesándolos con la función que corresponda y devolviendo el resultado a su nodo padre.

La Figura 28 muestra la estructura de un árbol. En esta figura se tienen diferentes símbolos que pueden ser ejecutados en cualquier orden. Pero existen convenciones para

la ejecución de estructuras de árboles. La ejecución convencional de la estructura de un árbol consiste en evaluar repetidamente el nodo que está más a la izquierda para todas las entradas disponibles. Este orden de ejecución es denominado. **orden postfijo**, llamado así por que los operadores aparecen después de los operandos. Otra ejecución convencional es el llamado **orden prefijo**, y se le llama así por que los operadores aparecen antes de los operandos. La forma más utilizada en GP es la prefijo que es la utilizada en LISP.

Aplicando el orden prefijo a la Figura 28, el orden de ejecución de los nodos es el siguiente: - (or (d, e), \* (a, + (b, c))).

### **Ejecución de la estructura lineal**

La gran mayoría de las computadoras ejecutan programas estructurados linealmente, utilizando instrucciones especiales para implementar sentencias de control. Además, aunado a la mejora en rendimiento lograda al utilizar instrucciones nativas, motivó el desarrollo de representaciones lineales en GP's (Poli *et al.* , 2001).

Una estructura lineal es simplemente una cadena de instrucciones que se ejecutan de izquierda a derecha, o bien, de arriba hacia abajo dependiendo de cómo se interprete dicha representación.

Un programa lineal empieza su ejecución en la instrucción del tope o de la izquierda, según esté representado y así continúa hacia abajo o hacia la derecha. Este tipo de representaciones permite ejecutar instrucciones de salto, por lo que los hace flexibles en su ejecución. Al finalizar la ejecución del programa representado en la Figura 29 produce el resultado almacenado en el registro a.

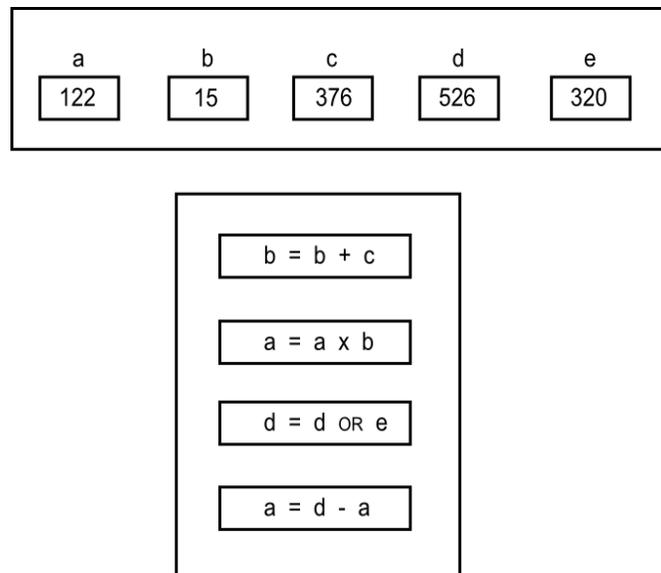


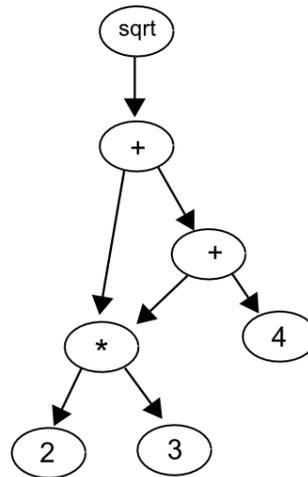
Figura 29: Estructura Lineal.

### Ejecución de la estructura de grafo

Siendo los árboles una clase particular de grafos, es natural evaluar el uso de representaciones mediante el uso de grafos más generales. Sin embargo estas aproximaciones no han alcanzado una difusión comparable a las dos anteriores. Los grafos son capaces de representar un programa muy complejo con estructuras compactas. La estructura de un grafo no es otra cosa que nodos conectados por aristas. Una arista es la conexión entre dos nodos, indicando la dirección del flujo del programa.

Las estructuras de árboles y programas lineales, son también grafos, sólo que éstos presentan restricciones particulares para la forma en cómo se conectan los nodos a través de las aristas. Una posible aplicación (Poli *et al.*, 2001) es la reutilización de los resultados de ciertas funciones. De esta forma, un grafo acíclico dirigido como en

la Figura 30 puede representar la reutilización de la subexpresión que se resalta. La utilización de un cache de expresiones previamente evaluadas disminuye la utilidad de estas aproximaciones, ya que disminuye los costos de evaluar repetidamente las mismas subexpresiones.



$(\text{sqrt} (+ (* \mathbf{2} \mathbf{3}) (+ 4 (* \mathbf{2} \mathbf{3}))))$

Figura 30: Grafo acíclico dirigido.

### III.3 Inicialización de la Población

El primer paso para llevar a cabo la ejecución de un programa en programación genética es inicializar la población. Esto significa crear una variedad de estructuras de programas para una evolución posterior. Por lo tanto, los árboles en programación genética, se definen por parámetros expresados tales como la máxima profundidad del árbol y el número máximo total de nodos en un árbol.

El *Parámetro de Máxima Profundidad (PMP)* es la profundidad más larga permitida entre el nodo raíz y los nodos terminales. En general, para los nodos de aridad 2, el tamaño del árbol tendrá un número máximo de  $2^{PMP}$  nodos. En la representación

lineal, el parámetro es llamado longitud máxima y se traduce en el número máximo de instrucciones permitidas en un programa. Para los grafos, el número máximo de un nodo es equivalente al tamaño del programa.

### III.3.1 Inicialización de las estructuras con árboles

Hay que recordar que los árboles son construidos por unidades llamadas funciones y terminales. Por ejemplo, supondremos que las terminales y funciones disponibles en los árboles del programa son las siguientes:

- $T = \{a, b, c, d, e\}$
- $F = \{+, -, x, \%\}$

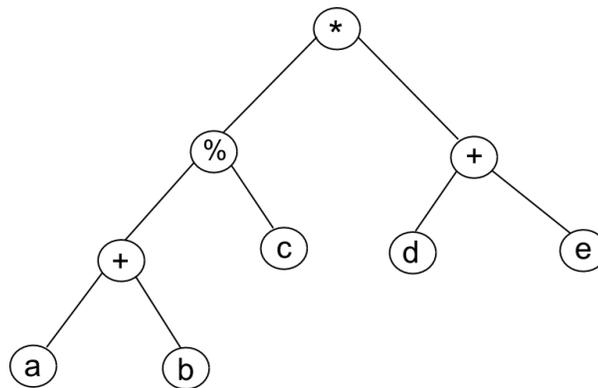


Figura 31: Individuo inicial utilizando con el método “grow” y con una PMP=4.

Existen dos métodos diferentes para la inicialización de las estructuras de los árboles:

- **Full:** Mediante este método los árboles tienen forma regular, es decir, es un árbol balanceado.

- **Grow:** Mediante este método los árboles tienen forma irregular.

En la Figura 31, la rama que termina con la entrada **d** tiene una profundidad de 3.

Con el método “Full”, se seleccionan nodos aleatorios desde el conjunto de terminales y funciones. En este método se seleccionan sólo funciones hasta el nodo que representa la máxima profundidad, y cuando se llega a este punto entonces se seleccionan sólo sus terminales. El resultado de este método es que cada rama del árbol cumple con la máxima profundidad.

En la Figura 32, se muestra un árbol que ha sido inicializado por el método completo con una profundidad máxima de 3.

### III.3.2 El método de mitad y mitad

La diversidad es valiosa en las poblaciones de la programación genética. Cuando se usa el método completo, puede ocurrir que se produzca un conjunto uniforme de estructuras en la población inicial ya que la rutina es la misma para todos los individuos. Para prevenir esto, se ha diseñado la técnica llamada “*mitad y mitad*” por sus siglas en inglés *Ramped Half-and-Half*.

La técnica consiste en lo siguiente: Supongamos que el parámetro de máxima profundidad es de 6. La población es dividida entre todos los individuos para ser inicializada con árboles que tengan profundidades de 2, 3, 4, 5 y 6. Para cada una de estas profundidades, la mitad de los árboles serían inicializados con la técnica completa y la otra mitad con la técnica que varía su forma (“*grow*”).

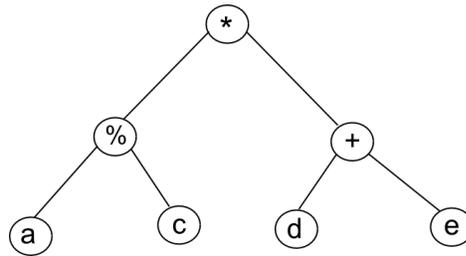


Figura 32: Individuo inicial utilizando el método “full” y PMP=3.

## III.4 Operadores Genéticos

Una vez que la población ha sido inicializada, usualmente la aptitud de la misma es muy baja pues estas soluciones son generadas al azar. A partir de esta población se procede a emular el proceso evolutivo transformando dicha población inicial con los operadores genéticos.

Entre los operadores podemos encontrar:

- Cruza,
- Mutación.

### III.4.1 Cruza

El operador genético denominado cruza combina el material genético de dos padres intercambiando un parte de uno de ellos con otra parte del otro. La cruza basada en árboles se muestra en la Figura 33, Los padres son mostrados en la parte superior, mientras que los hijos son los que están en la parte inferior. La cruza en los árboles funcionan de la manera siguiente:

- Se seleccionan dos individuos como padres, utilizando un mecanismo generalmente probabilístico. Los dos padres se muestran en la parte superior de la Figura 33.
- Se seleccionan aleatoriamente subárboles de cada padre. En la Figura 33 los subárboles seleccionados son mostrados con dobles círculos y con líneas más gruesas. La selección de los subárboles puede ser sesgada, de forma que los subárboles que sean terminales tengan baja probabilidad de ser seleccionados.
- Se intercambian los subárboles seleccionados entre los dos padres. El resultado de estos individuos son los hijos, los cuales se muestran en la Figura 33.

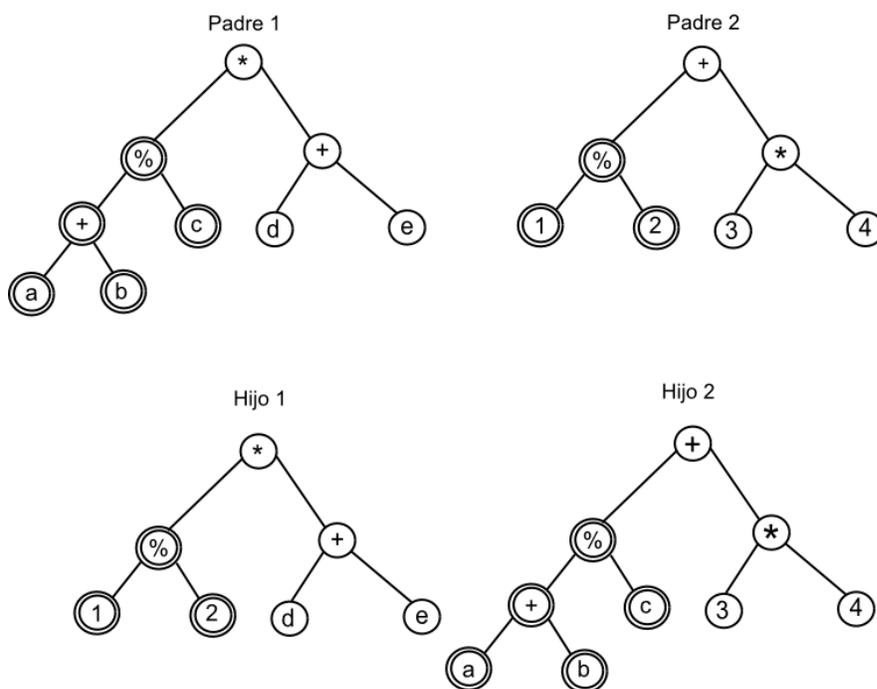


Figura 33: Máxima Profundidad de 3, inicializando con el método “full”.

### III.4.2 Mutación

El operador de mutación es aplicable a un solo individuo. Cada hijo producido por la cruce es sometido a mutación con una baja probabilidad definida por el usuario. Una aplicación separada de cruce y mutación es posible y puede resultar adecuada en algunos casos.

La mutación para árboles se muestran en la Figura 34. El operador genético denominado mutación reemplaza un subárbol por otro, pero cuidando que el nuevo individuo no exceda la profundidad máxima permitida.

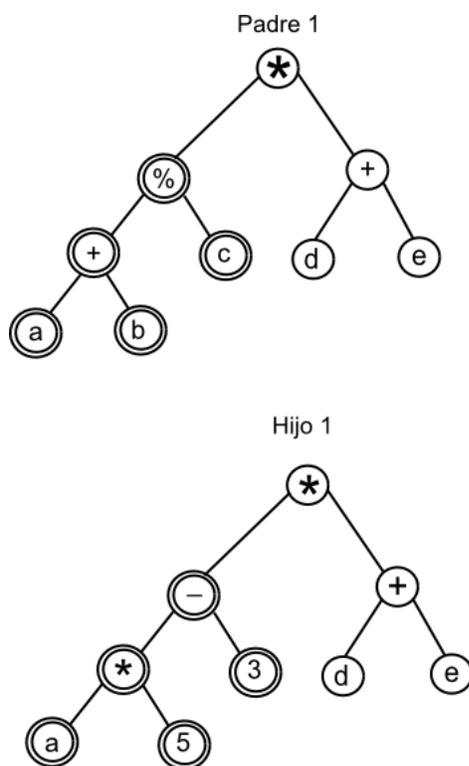


Figura 34: Mutación a un individuo.

La mutación aplicada a un individuo se realiza de la siguiente manera:

- El individuo es seleccionado de acuerdo a un porcentaje, el cual generalmente es bajo, El individuo original es el que se muestra en la parte superior, mientras que el resultado es el que se observa en la parte inferior de la Figura 34.
- Luego se selecciona aleatoriamente un subárbol del individuo. En la Figura 34 el subárbol seleccionado es mostrado con dobles círculos y con líneas más gruesas.
- Así, se reemplaza el subárbol seleccionado por otro que se crea aleatoriamente.
- Finalmente, se verifica la profundidad máxima permitida y si ésta es rebasada, entonces se procede de nuevo con la mutación. De lo contrario, el nuevo individuo será aquel que tenga el subárbol generado aleatoriamente.

## III.5 Aptitud y Selección

La programación genética es un tipo de búsqueda dirigida. Ésta debe seleccionar cuáles miembros de la población serán afectados por los operadores genético (cruza y mutación). Una vez que se ha realizado lo anterior, la GP implementa una de las partes más importantes del modelo de aprendizaje evolutivo, la selección basada en la aptitud del individuo. La métrica de evaluación en GP es llamada función de aptitud. Ésta juega el papel del medioambiente, determinando qué individuos sobrevivirán y cuáles no.

### III.5.1 Función de Aptitud

Una función de de aptitud (fitness en inglés) es un tipo particular de función objetivo que cuantifica cómo es de óptima una solución. La aptitud es la medida usada en la GP durante la simulación del proceso de evolución, y determina qué tan bien se ha aprendido el programa al predecir las salidas a partir de las entradas; en otras palabras,

determinar las características del dominio de aprendizaje. La evaluación de la función de aptitud tiene como objeto dar una retroalimentación al algoritmo de aprendizaje evolutivo identificando cuales individuos tendrán una mayor probabilidad de reproducirse. Asimismo, identificar cuáles individuos tendrán mayor probabilidad de ser removidos de la población.

Los problemas típicos que se resuelven de manera directa con la GP son por ejemplo los de regresión simbólica. En este tipo de problemas, se pretende generar una expresión matemática, combinando los operadores y las terminales disponibles que aproximen la curva definida por los datos proporcionados por el usuario. Algunos ejemplos de regresión simbólica son:

- Número de ejemplos clasificados correctamente en una tarea de clasificación.
- La desviación entre lo observado y la realidad en una aplicación de predicción.
- La cantidad de comida encontrada y consumida por un agente artificial en una aplicación de vida artificial.

### III.5.2 Algoritmo de Selección

Después de que se ha determinado la calidad de un individuo aplicando la función de aptitud, se decidirá si se aplican o no los operadores genéticos a un individuo y si se debe o no mantener en la población, a fin de permitir que éste sea reemplazado. Lo anterior se denomina, *Operador de Selección*.

Existen diferentes métodos de selección que es tarea del usuario en GP el decidir el método de selección que se aplicará tomando en cuenta las circunstancias específicas del problema. La selección es la responsable de la velocidad de evolución y de no

manejarse adecuadamente, puede provocar convergencia prematura. La selección en general, es una consecuencia de la competencia entre los individuos de una población.

### III.5.3 Algoritmo Básico de GP

Existen dos maneras de efectuar el reemplazo poblacional en la PG: el método generacional y el de estado uniforme no generacional. En el método generacional, la nueva población se forma a partir de la anterior y reemplaza totalmente a ésta.

En el método de estado uniforme, sólo unos pocos individuos son seleccionados y sujetos a cruce y mutación. Los descendientes producidos reemplazan a un número igual de la población, en su mayoría los peores. El concepto de generación no existe en este caso.

A continuación se dan los pasos preliminares de la GP:

- Definir el conjunto de las terminales.
- Definir el conjunto de funciones.
- Definir la función de aptitud.
- Definir los parámetros, tales como: el tamaño de la población, tamaño máximo del individuo, probabilidad de cruce, método de selección y el criterio de detención.

#### **Versión Generacional de la GP.**

1. Inicializar la población.
2. Evaluar a los individuos existentes en la población. Calcular la aptitud de cada individuo.

3. Hasta que la nueva población no esté completa, repetir los siguientes pasos:
  - Seleccionar un individuo o individuos en la población utilizando el algoritmo de selección elegido.
  - Aplicar los operadores genéticos sobre el individuo o individuos seleccionados.
  - Colocar a los descendientes producto de cruce y mutación en la nueva población.
4. Si el criterio de término se satisface se detiene el proceso; en caso contrario, reemplazar a la población existente con la nueva población y se repite los pasos 2-4.
5. Reportar al mejor individuo de la población.

#### **Versión de Estado Uniforme de la GP.**

1. Inicializar la población.
2. Seleccionar aleatoriamente un subconjunto de la población para que ésta tome parte en un torneo (competidores).
3. Evaluar la aptitud de cada competidor en el torneo.
4. Seleccionar al ganador o ganadores del torneo utilizando el algoritmo de selección.
5. Aplicar los operadores genéticos sobre el ganador o ganadores en el torneo.
6. Reemplazar a los perdedores del torneo con los resultado de la aplicación de los operadores genéticos aplicadas en los ganadores de los torneos.

7. Repetir los pasos 2-7 hasta que el criterio de paro sea completado.
8. Reportar al mejor individuo de la población

## III.6 Problemas resueltos exitosamente por la GP

Como todas las técnicas de resolución de problemas existentes, la programación genética se destaca particularmente en la solución de ciertas clases de problemas. A continuación se mencionarán algunas de las características de los problemas que han sido resueltos con éxito por la programación genética, siguiendo lo expuesto en los diversos trabajos del grupo de EvoVisión:

- **La interrelación de las variables no es conocida en una forma satisfactoria:** Cuando la forma en que se relacionan las variables es conocida con un cierto grado de detalle, pueden desarrollarse procedimientos analíticos con sus inherentes ventajas. Los procedimientos generales como la programación genética brillan en áreas en las que el conocimiento de las interrelaciones es incompleto.
- **Existe una forma rápida de evaluar soluciones tentativas al problema:** Como la programación genética debe evaluar un gran número de candidatos para poder obtener una solución razonable y efectiva, debe existir un procedimiento para evaluar soluciones parciales que puedan ejecutarse un gran número de veces con un costo computacional razonable.
- **Son aceptables soluciones parciales:** Al ser un procedimiento de búsqueda estocástica operando en general en un espacio enorme de posibles soluciones, la programación genética no puede garantizar encontrar un óptimo global para el problema. Por lo tanto, el problema debe poder aceptar soluciones no óptimas

ya que, de lo contrario, deberán emplearse en general técnicas analíticas para obtener la solución, siempre y cuando esto último sea posible.

- **No es necesario que los resultados tengan una justificación comprensible:** Como es normal en los procesos evolutivos, pueden producirse resultados que difieren notablemente de los que producirían un ingeniero humano en la misma situación. Si bien en algunos casos se han obtenido resultados muy similares a los encontrados previamente por seres humanos (Koza *et al.* , 2003), en otros los resultados del proceso evolutivo pueden resultar difíciles de comprender.

# Capítulo IV

## GPGPU

El cómputo de proposito general a la unidades de procesamiento gráfico (GPGPU<sup>1</sup>), es un concepto reciente dentro de las ciencias de la computación, que trata de estudiar y aprovechar las capacidades de computo de una unidad de procesamiento gráfico (GPU<sup>2</sup>), para cualquier ámbito. Las GPU's se han ido especializando durante los años en el proceso de vértices y píxeles, que básicamente acaba siendo multiplicaciones de vectores por matrices de números en coma flotante. Este tipo de cálculo, obviamente no es sólo característico del proceso de visualización 3D, si no que es muy habitual, especialmente en ámbitos científicos.

El alto poder de cómputo y la flexibilidad introducida en las recientes GPU's están permitiendo solucionar problemas, hasta hace poco desafiantes para PCs, usando hardware de bajo costo. Esto ha generado cambios importantes en todo el contexto ya que ha puesto al alcance hardware paralelo de muy alto rendimiento, que puede incluso ser utilizado en resolución de problemas no necesariamente del dominio del cómputo gráfico (Fung *et al.* , 2005), como en el caso de la visión por computadora.

Recientemente la capacidad de las tarjetas de video, han dejado de crecer linealmente para crecer exponencialmente, de manera específica las tarjetas a las cuales tiene

---

<sup>1</sup>GPGPU: del inglés General-Purpose Computing on Graphics Processing Units

<sup>2</sup>GPU: del inglés Graphics Processing Unit

acceso el consumidor común. Este tipo de tarjetas están avanzando mucho más rápido que la ley de More (esta ley dice que el poder de procesamiento del hardware se duplica cada 1.5 años). En los últimos cuatro años ha habido un avance tan drástico que es difícil mantenerse al día en cuanto al desarrollo tecnológico de las GPU y el poder de procesamiento de estas tarjetas. Las cuales se ha incrementado más de 10 veces en este periodo (Akenine-Möller , 2002). Las tarjetas de video que ahora están apareciendo en las computadoras de escritorio son casi tan poderosas como las que hay en las estaciones de trabajo de hace 3 años y en algunos casos de 1 año de antigüedad. Muestra de este avance, es que actualmente se generan imágenes, tracking en tiempo real que hace años tomaba horas y hasta días en hacerlo.

Se han realizado comparaciones hechas entre los CPU's de las computadoras y los GPUs de las tarjetas de video, ver Figura 35, en cuanto al número de instrucciones de punto flotante que pueden realizar en un segundo (FLOP/s). En ellas se pueden apreciar que los CPU's se han incrementado en promedio 2.0 veces su poder de procesamiento, mientras que los GPU's lo han incrementado en promedio 3.7 veces cada 18 meses (Wloka y Huddy , 2003).

## IV.1 GPU vs. CPU

Como se puede observar en el gráfico de la Figura 35, la evolución en los últimos años de las CPU's frente a las GPU's. Unos datos un poco más actuales, nos muestran que mientras un procesador de una computadora de gama media (*Core 2 Duo*) puede rondar los 10-15 GigaFlops; en contraste la capacidad de una tarjeta del mismo rango (*GeForce 9800*) puede alcanzar unos 420 GigaFlops.

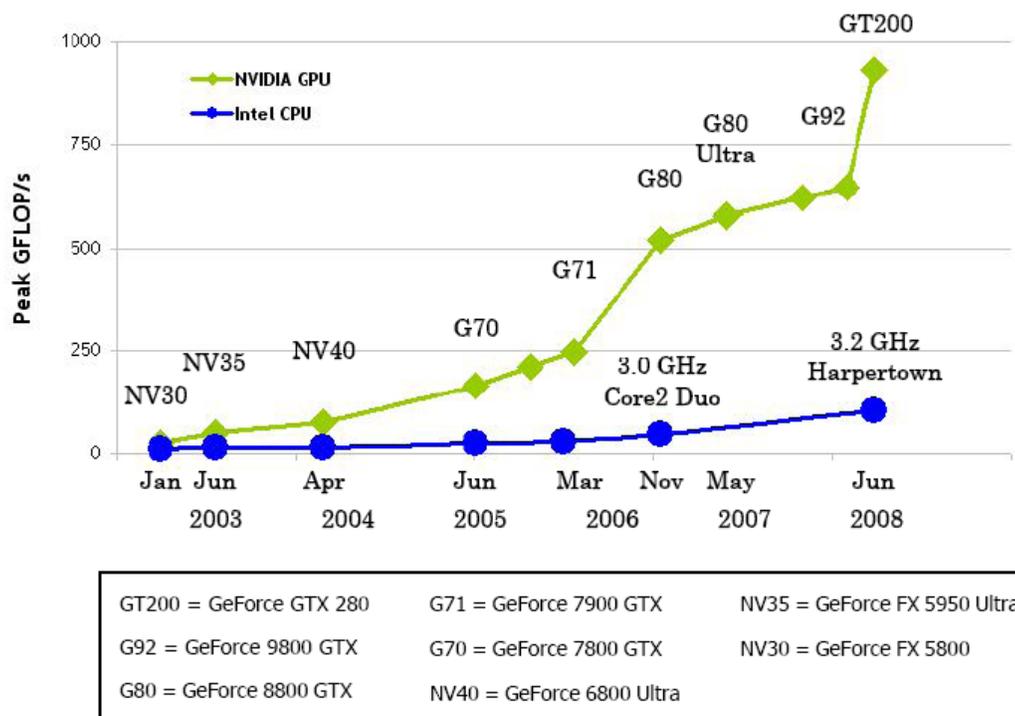


Figura 35: Comparación entre incremento de velocidad de los CPU's y los GPU's

Con las últimas actualizaciones, tanto de las tarjetas gráficas, así como de CPU's, el valor pico de las primeras se aproxima a los 500 GigaFlops y el de los segundos no ha aumentado más allá de los 20 GigaFlops. Esto se debe a las diferencias estructurales entre el CPU y el GPU, un aspecto importante es la velocidad del bus que comunica la GPU con la CPU. La capacidad del bus de datos para transferir información ya sea entre la CPU y la memoria residente en la placa base o entre la CPU y la GPU ha sido clave. Por desgracia el ancho de banda que son capaces de dar los buses de datos nunca han estado al nivel de las capacidades de computación de los microprocesadores, siempre han ido un paso por detrás. Esta situación ha sido un constante cuello de botella en

el mundo de los gráficos y por ello la conexión de la GPU se fue especializando y se colocó lo más cercana a la memoria y a la CPU, unido a una organización más eficiente de la memoria en capacidad de cálculo, ver Figura 36.



Figura 36: Diferencias entre la CPU y la GPU.

A pesar de que las tarjetas gráficas de la marca Nvidia han sido siempre muy bien acogidas dentro de la comunidad de jugadores, el mercado de la aceleración gráfica es demasiado voluble. Cuando *AMD* adquirió *ATI* en el año 2006, Nvidia se quedó como la única compañía independiente dedicada a la venta de GPU's ya que el resto de competidores habían caído por el camino (*Matrox*, *S3 Graphics*, *3dfx*, entre otros). Esta condición de superviviente debería ser envidiable, pero no lo fue tanto cuando se vislumbró que tanto *AMD* como *Intel* tenía planes de incluir núcleos gráficos en los futuros procesadores. Dichos procesadores, unidos a los computadores personales (especialmente destinados a empresas) que ya incluían una tarjeta gráfica integrada en la placa base, se harían con un porcentaje de mercado que podrían llegar a competir sensiblemente a Nvidia; sobre todo si se repetía el caso de los procesadores matemáticos, cuando fueron incluidos nativamente en los actuales microprocesadores.

## IV.2 CUDA

El deseo de poder utilizar toda la capacidad de cálculo para cualquier propósito ha llevado a desarrollar a: “*Compute Unified Device Architecture*”, comúnmente conocido por su acrónimo **CUDA**, es una plataforma de software para facilitar el cálculo paralelo desarrollado por la compañía *Nvidia* para sus GPU's, a partir de la serie 8 (en la gama GeForce). Presentado formalmente en 2006, tras una larga gestación de más de un año en modo beta, CUDA esta ganando clientes sin parar en el ámbito científico. Al mismo tiempo, Nvidia esta rediseñando el concepto de GPU y llevándolo más allá del de un simple dispositivo para juegos o gráficos 3D. De esa necesidad de diversificación y de la necesidad de aprovechar esa capacidad de cálculo descubierta con *Cg*, surgio está herramienta de programación.

CUDA es una tecnología muy reciente, pero en los últimos años se ha aplicado a numerosos campos consiguiendo considerables incrementos del rendimiento de algoritmos y aplicaciones como la búsqueda de los *k-nearest*, *neighbours*, *ray-tracing*, *dinámica de fluidos*, *protein-docking*, entre otros.

Retomando un poco la historia de CUDA, el cálculo paralelo en tarjetas gráficas no era algo nuevo para la compañía ya que desde principios de esta década, Nvidia venia desarrollando *Cg*. En lugar de escribir el código en ensamblador se usa un lenguaje de más alto nivel similar al C, pero a diferencia de CUDA, *Cg* se dirige específicamente a la generación de gráficos. A medida que el número de núcleos se fue incrementado en una GPU y por ende ganando en rendimiento al operar con datos en coma flotante; se hizo cada vez más patente la necesidad de aprovechar esa potencia en cálculos para



Figura 37: Nvidia CUDA

propósitos alternativos, aparte de la evidente aplicación de videojuegos de última generación.

CUDA, aunque es una tecnología muy joven, está bastante bien documentada por parte de sus creadores, NVIDIA, y existen documentos y manuales disponibles. Además, su popularidad en el mundo de la programación en general y en especial en el mundo académico, ha ocasionado la aparición de tutoriales, cursos online y cursos en varias universidades de todo el mundo dedicados a conocer y desarrollar programas en CUDA.

#### **IV.2.1 Arquitectura CUDA**

CUDA es un modelo de programación paralelo y un entorno de programación que tiene por objetivo facilitar el desarrollo de aplicaciones que puedan ejecutarse en los multiprocesadores de las tarjetas gráficas, sin tener que preocuparse por las características de cada modelo de tarjeta

## Modelo de ejecución

En CUDA la unidad básica de ejecución son las hebras ligeras (*threads*), es decir, es una serie de instrucciones de código que se ejecutan de forma secuencial. Los threads se ejecutan en una serie de *Streaming Multiprocesors* (multiprocesadores) capaces de ejecutar muchos threads simultáneamente. Este tipo de procesadores reciben ese nombre porque implementan el paradigma del *stream processing*.

*Stream* se podría interpretar en este contexto como un arroyo y el símil está en que en esta forma de proceso de datos hay un flujo o arroyo de datos al que se le aplican una serie de operaciones. La clave está en que las operaciones siempre son las mismas, por lo que son tareas fácilmente paralelizables. Las primeras versiones de procesadores capaces de efectuar estos cálculos las operaciones estaban implementadas en hardware y eran fijas, sin embargo en las GPUs actuales hay suficiente libertad para efectuar casi cualquier tipo de cálculo.

Otra forma de denominar este paradigma de computación es *SIMD* (*Single Instruction Multiple Data*), aunque los desarrolladores de NVIDIA han denominado *SIMT* (*Single Instruction Multiple Thread*) a su arquitectura, ya permite a los threads que se ejecutan en paralelo llevar varios caminos (branch) a diferencia de los procesadores vectoriales que no permiten ninguna diferencia en las ejecuciones de las instrucciones.

Entrando más en detalles de la implementación de CUDA hay que explicar que cada multiprocesador está compuesto por una serie de procesadores escalares, por una memoria que comparten todos los *threads* que se están ejecutando en el multiprocesador, una caché para la memoria de textura, una caché para la memoria constante y una lógica capaz de crear y planificar los threads con muy poco coste.

Los threads se agrupan en warps, un warp no es más que un grupo de 32 *threads* que se ejecutan en paralelo. Todos los threads de un warp empiezan a ejecutarse al mismo tiempo y siguiendo las mismas instrucciones de código, pero pueden tomar caminos diferentes. En realidad se ejecuta primero un *half-warp* (la mitad de un warp) y después la otra mitad, es un dato importante a la hora de conseguir accesos eficientes a memoria o el mínimo de threads divergentes (que ejecutan código diferente al del resto del half-warp). La divergencia de threads es algo a evitar ya que cuando un thread toma un camino de ejecución diferente al del resto del *half-warp*, todos los *threads* que no siguen ese camino se bloquean y esperan hasta que todos los threads converjan en la misma instrucción.

En un nivel superior los *threads* se agrupan en bloques y todos los bloques de un *kernel* en un *grid* o cuadrícula. Cada bloque es asignado a un multiprocesador, y éste es el encargado de dividir los threads del bloque en *warps*. Los divide siempre de la misma forma, siguiendo el orden de sus identificadores, la asignación de identificadores a threads se explica más adelante.

La cantidad de bloques que se pueden asignar a un multiprocesador depende de la cantidad de registros y memoria compartida que necesiten los threads que lo componen. Por el diseño de los multiprocesadores puede haber un máximo de 8 bloques activos por multiprocesador y si se da el caso que un bloque por si solo necesite más cantidad de registros o memoria compartida de la que disponen los multiprocesadores se producirá un error.

## Hardware

La arquitectura de una GPU actual en Nvidia tiene aproximadamente el siguiente aspecto, ver Figura 38. En ella se puede ver como los multiprocesadores (en verde) tienen como entrada el código de los diferentes tipos de *shaders* (*vertex*, *geom* y *pixel*). En azul está representada la memoria de textura y en naranja la memoria de la GPU.

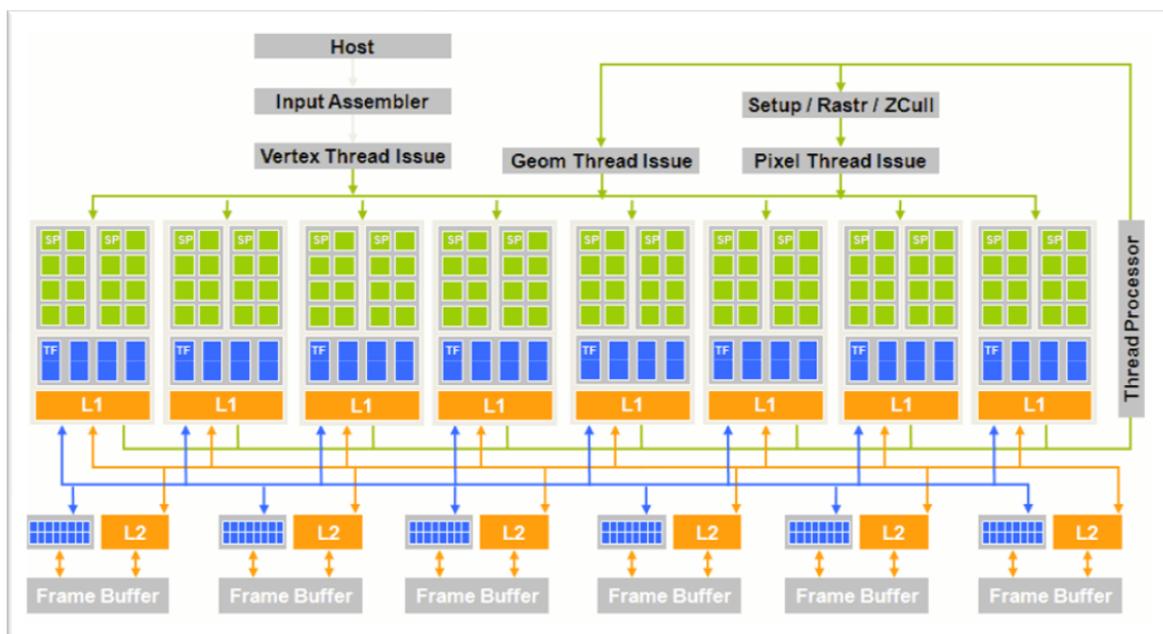


Figura 38: Esquema del hardware de una GPU.

Los multiprocesadores son las unidades de cálculo que hacen todo el trabajo de computación. Se componen de varios núcleos (24 en las versiones actuales) y son capaces de ejecutar de forma paralela un thread en cada núcleo, siempre que todos los *threads* ejecuten las mismas instrucciones. Una GPU puede tener más de un multiprocesador, de manera que su capacidad de cálculo se incrementa al ser capaz de ejecutar más

threads de forma simultánea. Una de las principales ventajas de CUDA es que el escalado de multiprocesadores es totalmente transparente para el programador ya que los gestores de threads se ocupan automáticamente de distribuir y sincronizar a los threads en los multiprocesadores.

Las tarjetas gráficas que soportan CUDA son capaces de usar estos circuitos de manera ligeramente distinta respecto a cuándo llevan a cabo el proceso de visualización. El esquema del hardware se representa en la Figura 39.

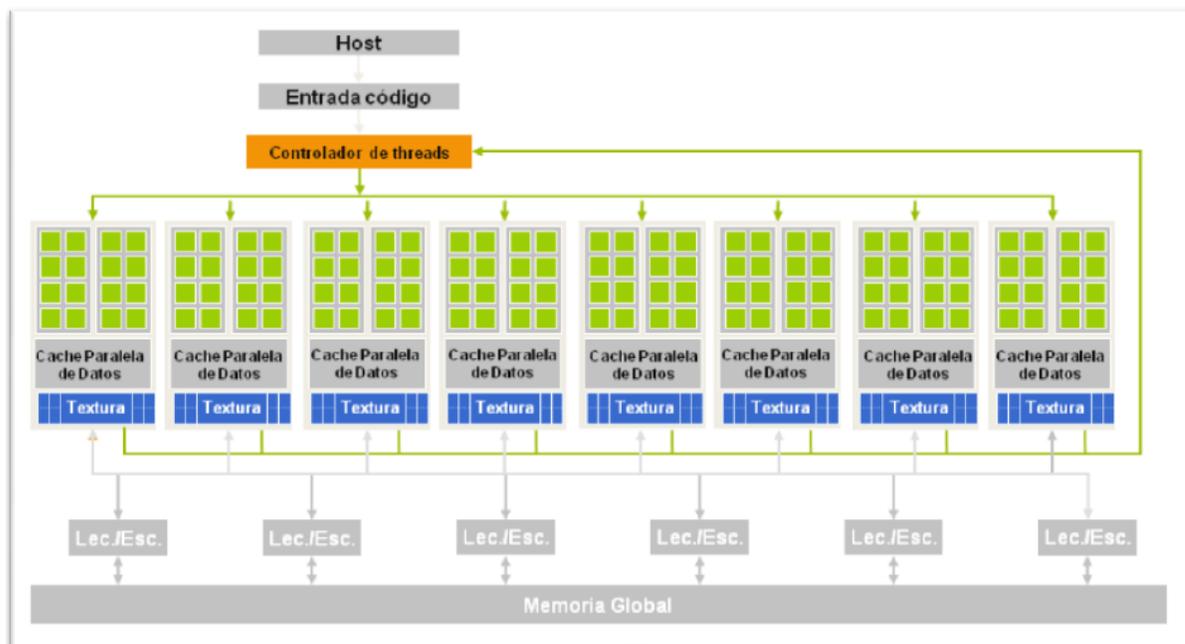


Figura 39: Esquema de una GPU cuando ejecuta aplicaciones en CUDA.

El esquema se simplifica respecto al original ya que la entrada de datos se reduce a un solo tipo y ya no se hacen escrituras en el *Frame Buffer*. Además la estructura de memoria se simplifica de forma que el nivel de cachés se reduce y sólo existe caché para la memoria constante y la de textura pero no para la memoria global.

## Estructura de la memoria

Existen varios niveles de memoria en la arquitectura de CUDA. La memoria más cercana a la unidad de proceso son los registros y la memoria compartida, justo después nos encontramos con las caché de datos de memoria constante y de textura, que también se encuentran dentro del multiprocesador. Ubicadas fuera del multiprocesador tenemos la memoria global en la que se puede leer o escribir, la memoria constante y la memoria de textura, estas dos últimas sólo de lectura. En un nivel intermedio nos encontramos con la memoria local, que es accesible tanto de lectura como de escritura a nivel de thread pero se encuentra alojada en memoria global, esta memoria se usa sobre todo para grandes estructuras de datos que no pueden ser alojadas en registros o memoria compartida. Otra memoria presente en los multiprocesadores pero invisible al programador es la memoria de instrucción que se encuentra fuera del multiprocesador, pero está en caché y su tiempo de acceso es pequeño. Desde la CPU o host sólo se puede acceder a la memoria constante, de textura o a la global.

El tiempo de acceso varía considerablemente de un tipo de memoria a otro. La memoria que se encuentra más cercana a la unidad de proceso es lógicamente más rápida mientras que la memoria más alejada es mucho más lenta.

Como podemos observar en la tabla anterior el acceso a memoria global es varios órdenes de magnitud superior al de otros tipos de accesos, por lo que una estrategia básica para mejorar el rendimiento de las aplicaciones es reducir al máximo el acceso a memoria global y en el caso de que los accesos sean imprescindibles aprovechar en todo lo posible los recursos ofrecidos por CUDA para mejorar el ancho de banda, como el correcto alineamiento de las estructuras de datos o los accesos coalesced o agrupados que se explican más adelante.

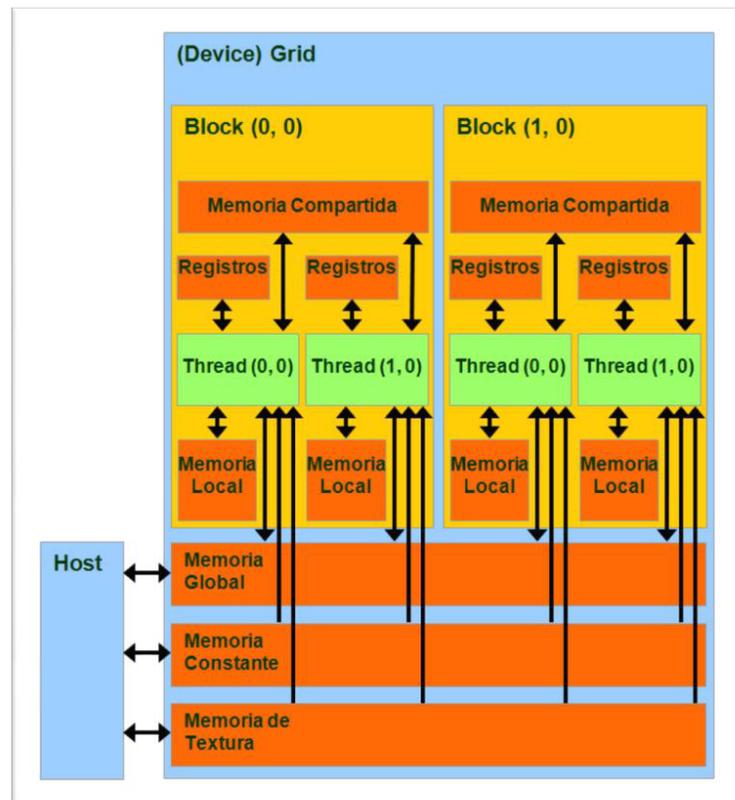


Figura 40: Estructura de la memoria en la GPU.

## IV.2.2 Estructura de un programa en CUDA

Tal y como se muestra en la Figura 42, durante la ejecución de un programa en CUDA se intercalan o incluso se superponen ejecuciones de código en la CPU o *host* y en la GPU o *device*. La CPU se suele encargar de la inicialización de los datos, de las transferencias de datos a la memoria de la GPU, del lanzamiento de los kernels y de recoger sus resultados, mientras que la GPU hace los cálculos.

Tipo de memoria		Número de ciclos
Registro		1
Memoria Compartida		1
Memoria Constante	En caché	1 - 10
	No en caché	10 - 100
Memoria Textura	En caché	1 - 10
	No en caché	10 - 100
Memoria Global / Local		400 - 600
Memoria de Instrucción		4

Figura 41: Estructura de la memoria en la GPU.

### IV.2.3 Threads, blocks y grids

Una GPU puede gestionar miles o incluso millones de *threads* de forma simultánea, pero para facilitarle el trabajo se pide al programador que agrupe los *threads* que va a lanzar en bloques y a su vez los bloques en un *grid* o cuadrícula.

La forma en que se organizan los *threads* en bloques y después los bloques en el grid es importante tanto para facilitar la programación del algoritmo como para conseguir un programa eficiente. En la figura anterior se puede observar cómo se numeran los *threads* y los bloques. Esta numeración es accesible durante la ejecución de los threads

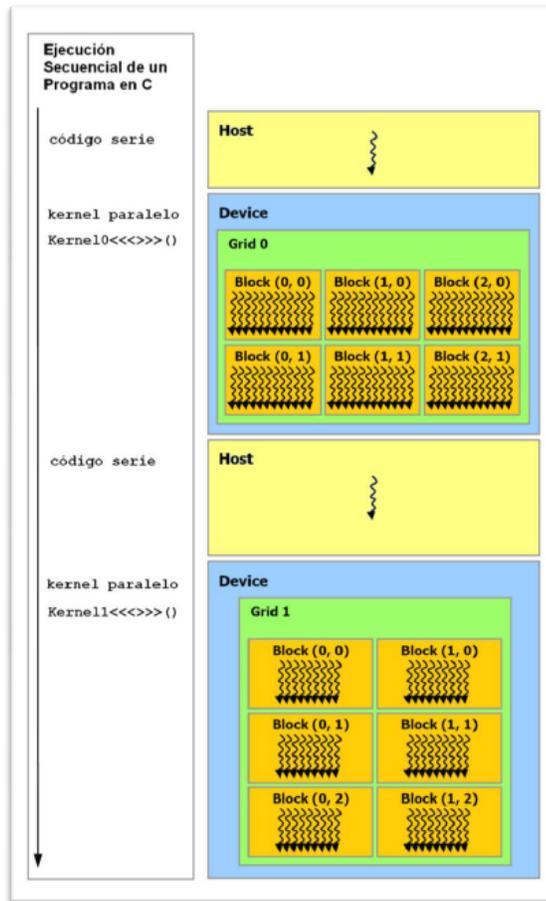


Figura 42: Estructura de la ejecución de un programa en CUDA.

permitiendo identificar de forma inequívoca que *thread* de que bloque se está ejecutando. El uso más habitual de estos identificadores es la indexación de memoria para conseguir los datos que corresponde tratar a ese *thread*.

El orden de ejecución de los *threads*, es decir, cómo se agrupan en warps para su ejecución es un recorrido por filas tanto de los bloques del grid como de los *threads* dentro del bloque. Si el bloque tiene tres dimensiones el recorrido para la ejecución de los *threads* pasa primero por las filas de los threads ubicados a menor  $z$ .

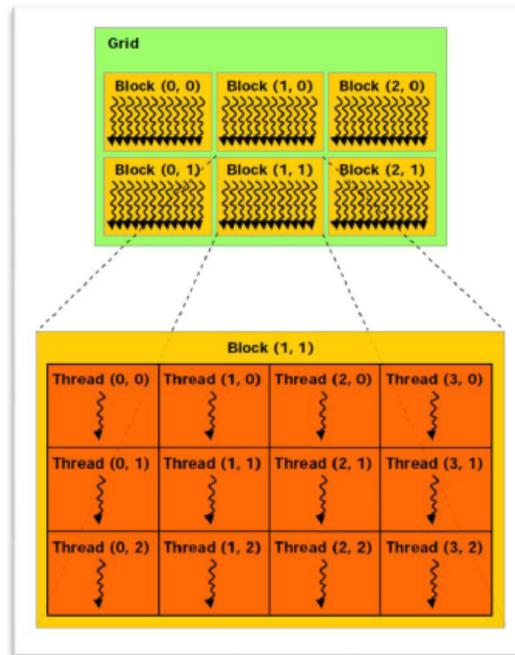


Figura 43: Threads, bloques y grids.

Una característica muy importante de CUDA es que todos los *threads* de un bloque comparten memoria dentro del multiprocesador, la llamada memoria compartida o *shared memory*. Es una herramienta importante a la hora de conseguir un buen rendimiento ya que es el lugar perfecto para almacenar cálculos intermedios y reducir los accesos a memoria global. Además los *threads* se pueden sincronizar a nivel de bloque, por lo que es importante tomar buenas decisiones en el momento de distribuir los *threads* en bloques. Una vez distribuidos los threads en bloques el paso de distribuir los bloques en el grid es algo más directo y depende de las dimensiones de los datos. Un factor que limita algo la libertad del programador es que mientras que los bloques pueden ser una matriz de hasta tres dimensiones, un grid sólo puede ser una matriz de hasta dos dimensiones.

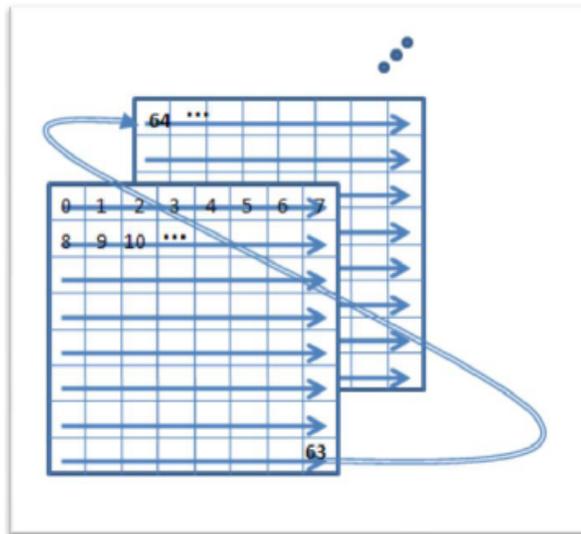


Figura 44: Ejemplo orden de ejecución de los threads de un bloque.

#### IV.2.4 Concurrencia CPU-GPU

La CPU y la GPU son unidades separadas e independientes, por lo que no tienen ningún problema en trabajar en paralelo. Es algo muy habitual durante el proceso de visualización, mientras la tarjeta gráfica se encarga de los cálculos necesarios para la visualización, mientras la CPU sigue trabajando. Con CUDA se ha mantenido esa independencia.

Cuando la aplicación lanza un *kernel* se retorna de inmediato el control al código de la CPU a no ser que se pida explícitamente usando la interfaz de CUDA, hay funciones que fuerzan a que todos los *threads* en ejecución en la GPU hayan terminado para que se devuelva el control a la CPU. Otros eventos bloqueantes son las transferencias de memoria entre CPU y GPU.

Para gestionar esa concurrencia en las aplicaciones CUDA ofrece unos elementos llamados *streams*. Un *stream* en CUDA es una secuencia de operaciones que se ejecutan

en orden. Es muy útil cuando se lanzan varios kernels y se quiere garantizar el orden en que se ejecutan sin necesidad de que intervenga la CPU. CUDA ofrece funciones para gestionar y sincronizar los diferentes *streams* que se estén ejecutando y también permite bloquear toda concurrencia.

# Capítulo V

## Reconocimiento de objetos

La visión por computadora se puede considerar como el conjunto de técnicas y modelos que permiten el procesamiento, análisis y la interpretación de la información considerada relevante. Por lo tanto, el objetivo principal es: “Proveer del sentido de la vista a sistemas autónomos para que puedan interactuar de forma eficiente en ambientes complejos (Branch y Olague , 2001).

El reconocimiento de objetos dentro de un sistema de visión por computadora está compuesta de varias etapas, ver Figura 45. En la primera se encuentra la adquisición de la imagen, como segunda etapa está el procesamiento de la imagen, otra etapa se encarga del análisis de la imagen y por último el de visión artificial. Cabe destacar, que dependiendo del sistema de visión a implementar pueden surgir variantes de estas etapas. A continuación se describen brevemente estas etapas:

1. En *adquisición* de la imagen se involucran modelos de la cámara que genera la imagen y varios principios geométricos en su captura.
2. En la etapa de *procesamiento* de la imagen es necesario eliminar información irrelevante, por ejemplo el ruido o en algunos otros como reconstruyendo imágenes podemos eliminar información irrelevante, aplicando operaciones de procesamiento de imágenes básicos como son los filtros, comprensión, segmentación, entre algunos otros, y

3. En la etapa del *análisis*, como la extracción de la información relevante a través de características locales ó puntos de interes en la imagen.

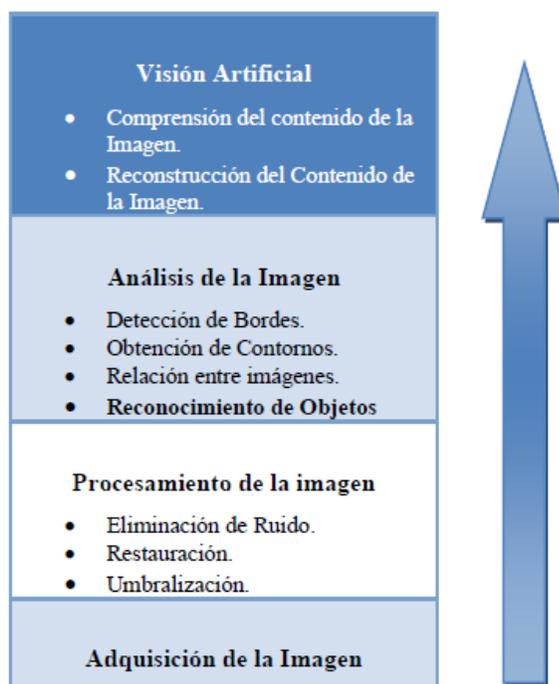


Figura 45: Etapas que conforman la visión por computadora.

Una vez establecido en el contexto, se puede definir el reconocimiento de objetos como: el proceso de identificación de objetos comparando ciertas características extraídas por los sensores utilizados, para percibir las características previamente almacenadas de varios objetos (Roy *et al.* , 2004).

El reconocimiento de objetos es un trabajo de visión por computadora que lleva cerca de 50 años de investigación, sobre el que se ha invertido gran esfuerzo debido a su amplitud de posibilidades prácticas. Los sistemas de visión por computadora utilizan

las técnicas de reconocimiento de imágenes para interpretar las propiedades de una imagen, compararla con su base de datos e identificar un objeto conocido. Esta tecnología y otros métodos básicos de procesamiento de imágenes se aplican actualmente en tareas como identificación de caracteres, diagnóstico médico, tecnología del espacio, detección de fisuras en la producción de vidrio, tratamiento de residuos sólidos urbanos, reconocimiento de huellas digitales, la clasificación de cuerpos celestes, además de otras relacionadas con la robótica, facilitando algunas tareas tales como: la interacción de éstos con su entorno, el comportamiento de sus movimientos y recorridos. En otras palabras tiene como fin extraer propiedades del mundo a partir de un conjunto de imágenes.

En la actualidad se comienza a generar los medios necesarios para emular de manera muy aproximada el comportamiento humano a fin de realizar actividades específicas de la vida cotidiana. En particular las personas realizamos esta interacción a través de los sentidos. En el caso del sentido de la vista, para poder interactuar con el entorno, el sistema visual humano debe ser capaz de reconocer los diferentes objetos que aparecen en nuestro campo visual.

En este trabajo, se tienen a las imágenes agrupadas en clases y se considera que para identificar dicha imagen se requiere de una descripción de la mismas. Con esta información se propone un sistema que permite reconocer e identificar el objeto para su correcta clasificación hacia la clase de objetos correspondiente. Así, la idea es alcanzar una solución al problema de la visión artificial aunque sea parcial, definiendo para ello dos etapas:

1. Un descriptor, que utilizaremos para describir tanto a los objetos como a la escena que los contenga.

2. Un algoritmo clasificador que, gracias a las descripciones de diferentes objetos, nos indique si otra descripción forma parte de nuestra colección de objetos.

## V.1 Descripción y extracción de características

Para reconocer un objeto es necesario tener una descripción del mismo (descriptor o modelo del objeto). Los descriptores deben ser independientes del tamaño, su localización y orientación del objeto; además, deben ser suficientes para discriminar objetos entre sí. Los descriptores se basan en la evaluación de algunas características del objeto, por ejemplo:

- a) Descriptores unidimensionales: códigos de cadena, forma del perímetro.
- b) Descriptores bidimensionales: área, momentos de inercia, entre otros.
- c) Descriptores específicos: rasgos diferenciadores de un objeto.

Uno de los principales problemas en el reconocimiento de objetos, es encontrar una manera óptima de representar la información original que describe a cada uno de los patrones<sup>1</sup> basados en los descriptores mencionados inicialmente. Este problema es conocido como extracción de características. Este proceso de *extracción de características* trata de reducir la cantidad de información (reducción de la dimensionalidad) que representa a cada uno de los patrones, obteniendo de esta forma, un *vector de características*<sup>2</sup> que represente de la mejor manera posible al objeto.

---

<sup>1</sup>Un patrón es una descripción cuantitativa o estructural de un objeto o alguna entidad de interés.

<sup>2</sup>Composición de varias características, está contiene la medida de las características de un patrón; puede estar formado de números binarios o valores reales. Un *vector de características* define puntos en un espacio n-dimensional.

### V.1.1 Requisitos en la extracción de características

La *extracción de características*, se refiere al proceso de obtener algunas medidas numéricas e información relevante de los datos que se desean analizar. Conviene destacar que las propiedades más importantes que deben seleccionarse durante la extracción de características, son:

- a) Discriminación: valores numéricos diferentes para objetos de clases<sup>3</sup> diferentes.
- b) Fiabilidad: cambios numéricos pequeños para objetos de una misma clase, es decir los objetos de una misma clase deberán representar la menor dispersión.
- c) Incorrelación: la independencia de las características equivale al principio de la parsimonia; decir lo mismo con la máxima economía de términos. Nunca debe utilizarse características que dependen fuertemente entre sí, ya que no añaden información. Se ha de tener la máxima información con el mínimo número de características.
- d) Invarianza: frente a transformaciones geométricas como rotaciones, translaciones y escalamientos.
- e) Dimensionalidad: el tamaño del vector de características deber ser menor que la del patrón original. Las características deben representar una codificación óptima de la entrada, desechando información que no sea importante, pero que refleje lo esencial del objeto.
- f) Cálculo en tiempo real: éste es un requisito que puede llegar a ser determinante en ciertas aplicaciones en tiempo real, ya que las características deben calcularse en un tiempo aceptable.

---

<sup>3</sup>Una clase es un conjunto de patrones que comparten algunas propiedades.

## V.2 Clasificación

Un clasificador es un método para determinar la posible clase de un objeto desconocido, o evento sobre la base de un número de casos de cada una de las clases, conocido como el conjunto de entrenamiento.

El primer paso en la clasificación es la extracción de características, que es donde cada instancia en el conjunto de entrenamiento se expresa como un vector de medidas. Cuando están siendo clasificadas las imágenes, este vector podría estar definido a partir de las intensidades de los píxeles, pero con frecuencia se aplica un paso de reducción de características. Las medidas que se observan de las características pueden ser números reales, enteros ó categorías las cuales denominan el espacio de características. Para algunos problemas no todas las mediciones están disponibles y esto se conoce como datos faltantes.

Una vez que las características se han extraído hay dos posibles casos, que en general se manejan de forma muy diferente. En el primer caso, la clase de cada instancia del conjunto de entrenamiento se pone a disposición del clasificador. En el segundo caso, la información no está disponible y esto se llama clasificación sin supervisión o “clustering”.

### V.2.1 Clasificación supervisada

Clasificación supervisada es donde la clase de cada una de las instancias en el conjunto de entrenamiento es conocida. Los clasificadores supervisados más utilizados son los clasificadores binarios, que distinguen entre dos tipos de objetos o evento. La función

de clasificación se conoce como discriminante. En el caso de los clasificadores multi-clase estos se generan frecuentemente a partir de un número de clasificadores binarios combinados. El rendimiento de un clasificador supervisado tiene como salida la estimación de la clase más probable dado un espacio de características. La medida obvia del rendimiento, se realiza al comparar la estimación de la clase contra la otra clase. El error del clasificador suele venir dado por el número total de casos mal clasificados.

El rendimiento real de un clasificador viene dado por las clasificaciones realizadas sobre datos que nunca ha visto antes. Mejorar el rendimiento del conjunto de entrenamiento puede tener aparentemente un efecto paradójico de empeorar la clasificación en otros datos (un efecto que se conoce como sobreentrenamiento). Por esta razón, al evaluar el rendimiento de un clasificador, todos los datos disponibles se dividen en tres conjuntos distintos. El primero es el conjunto de entrenamiento, el segundo es un conjunto de validación que se utiliza para dar una medida intermedia del rendimiento del clasificador que tiene un conjunto de los parámetros que hay que ajustar. El conjunto final es un conjunto de prueba, que se utiliza para evaluar el rendimiento del clasificador ajustado.

Tipos de clasificadores supervisados:

1. Vecinos más cercanos
2. Analisis lineal discriminante
3. AdaBoost
4. Máquinas de Soporte Vectorial

### Vecino k-más cercano

*K-nn* (*K nearest neighbors*) (Fix y Hodges , 1951) es un método de clasificación supervisada (Aprendizaje ó estimación basada en un conjunto de entrenamiento y prototipos) que sirve para estimar la función de densidad  $F(x / C_j)$  de las predicciones  $x$  por cada clase  $c_j$ . Este es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento  $x$  pertenece a la clase  $C_j$  a partir de la información proporcionada por el conjunto de prototipos.

A continuación se explica brevemente el algoritmo: los ejemplos de entrenamiento son vectores en un espacio característico multidimensional, donde cada ejemplo está descrito en términos de  $p$  atributos considerando  $a$  clases para la clasificación. Los valores de los atributos del  $i$ -ésimo ejemplo (donde  $1 \leq i \leq n$ ) se representan por el vector  $p$ -dimensional:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ip}) \in X \quad (1)$$

El espacio es particionado en regiones definidas por localizaciones y etiquetas de los ejemplos de entrenamiento. Un punto en el espacio es asignado a la clase  $C$  si esta es la clase más frecuente entre los  $k$  ejemplos de entrenamiento más cercano. Generalmente se usa la distancia euclideana:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^p (x_{ir} - x_{jr})^2} \quad (2)$$

aunque otras medidas de distancia, tales como Distancia de Manhattan podría en un principio ser utilizadas en lugar de la distancia euclideana.  $k - nn$  el algoritmo vecino más cercano es sensible a la estructura local de los datos.

La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, la evaluación del ejemplo (del que no se conoce su clase) (ver Figura 46) es representada por un vector en el espacio característico. Para esto se calcula la distancia entre los vectores almacenados y el nuevo vector, y se seleccionan los  $k$  ejemplos más cercanos. El nuevo ejemplo es clasificado con la clase que más se repite en los vectores seleccionados. Este método supone que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos. El problema de dicha suposición es que es posible que se tengan muchos atributos irrelevantes que dominen sobre la clasificación: dos atributos relevantes perderían peso entre otros veinte irrelevantes. Para corregir el posible riesgo se puede asignar un peso a las distancias de cada atributo, dándole así mayor importancia a los atributos más relevantes. Otra posibilidad consiste en tratar de determinar o ajustar los pesos con ejemplos conocidos de entrenamiento. Finalmente, antes de asignar pesos es recomendable identificar y eliminar los atributos que se consideran irrelevantes.

La selección de  $k$  depende de los datos; generalmente, valores más grandes  $k$  reducen el efecto del ruido en la clasificación, pero hace los límites entre las clases menos distintivos. Un buen  $k$  puede ser seleccionado por varias heurísticas, por ejemplo validación cruzada. El caso especial donde la clase se predice para ser la clase de la muestra más cercana del entrenamiento (es decir, cuando  $k = 1$ ) se llama el algoritmo del vecino más cercano.

La exactitud de  $k - nn$  se puede degradar seriamente por la presencia de características ruidosas o inaplicables, o si las escalas de las características no son constantes

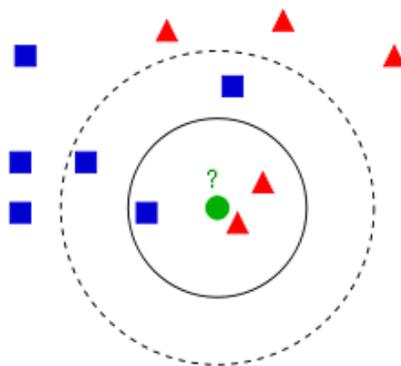


Figura 46: Ejemplo del algoritmo  $K - nn$ . Se desea clasificar el círculo verde. Para  $k=3$  este es clasificado con la **clase triángulo**, ya que existe solo un cuadrado y 2 triángulos, dentro del círculo que los contiene. Si  $k=5$  este es clasificado con la **clase cuadrado**, ya que hay 2 triángulos y 3 cuadrados, dentro del círculo externo.

con su importancia. Mucho esfuerzo en investigación se ha invertido en el problema de seleccionar características para mejorar la clasificación. Un acercamiento particularmente popular, es el uso de algoritmos evolutivos para optimizar el escalamiento de características.

### Análisis lineal discriminante

LDA o *Linear Discriminant Analysis* es una técnica de aprendizaje supervisado para clasificar datos. La idea central de LDA es obtener una proyección de los datos en un espacio de menor ó incluso de igual dimensión que los datos entrantes, con el fin de que la separabilidad de las clases sea la mayor posible. Es una técnica supervisada ya que para poder buscar esas proyecciones se debe entrenar el sistema con patrones etiquetados.

Existen varias implementaciones de LDA, entre ellas se encuentra Fisher-LDA. Para explicarlo vamos a considerar la versión más simple del problema:

*Encontrar el vector  $w$  de proyección, que proyecte los datos a un espacio uni-dimensional de manera de obtener la mayor separabilidad entre sus clases.*

Formalizando, tenemos  $x_1..x_n$  patrones  $d$ -dimensionales etiquetados en  $c$  clases. Cada clase cuenta con  $N_c$  patrones. Se busca  $w$  para obtener  $y_i = w^T x_i$  proyecciones uni-dimensionales de los patrones.

Lo que busca **Fisher-LDA** es maximizar la siguiente función objetivo:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \quad (3)$$

donde  $S_B$  es la matriz de dispersión inter-clase y  $S_W$  es la matriz de dispersión intra-clase. Siendo más precisos:

$$S_B = \sum_c N_c (\mu_c - \mu)(\mu_c - \mu)^T \quad (4)$$

$$S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \quad (5)$$

Siendo  $\mu_c$  la medida de cada clase,  $\mu$  la media de todos los datos,  $N_c$  la cantidad de patrones de la clase  $c$ .

Fisher-LDA busca encontrar el vector  $w$  de proyección que maximice el “cociente” entre la matriz de dispersión inter-clase y la matriz de dispersión intra-clase.

Operando se puede ver que  $w$  maximiza la función objetivo la cual debe cumplir:

$$S_B w = \lambda S_W w \quad (6)$$

Si  $S_w$  es no-singular podemos resolver el clásico problema de valores propios para la matriz  $S_W^{-1}S_B$ :

$$S_W^{-1}S_B W = \lambda w \quad (7)$$

Si ahora sustituimos la solución en forma analítica obtenemos lo siguiente:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} = \lambda_k \frac{w_k^T S_B w_k}{w_k^T S_W w_k} = \lambda_k \text{ con } k = 1..d \quad (8)$$

siendo  $w_k$  el vector propio de  $k$  de valor propio  $\lambda_k$ .

En consecuencia para maximizar la solución debemos considerar el vector propio con mayor valor propio asociado.

Claro está que este desarrollo valió para el caso en que queremos proyectar los datos sobre un espacio uni-dimensional. Se puede ver sin mayor esfuerzo (Duda , 2000) que para el caso de querer proyectar sobre un espacio m-dimensional, se debe resolver el mismo problema y elegir los  $m$  vectores propios asociados más grandes.

## AdaBoost

*Adaboost* es un algoritmo de aprendizaje formulado por Yoav Freund y Robert Schapier, es un algoritmo utilizado para construir clasificadores sólidos utilizando combinación lineal de clasificadores simples. Este algoritmo es adaptable, en el sentido de que la creación de los siguientes clasificadores se ajusta a favor de las instancias clasificadas erróneamente por los anteriores clasificadores.

La agrupación de varios clasificadores, obtenidos utilizando un mismo método, es

una manera natural de incrementar la precisión, con respecto a la obtenida con la utilización aislada de dichos clasificadores. Uno de los métodos más populares para crear estas agrupaciones de clasificadores es el boosting. Este término engloba toda una familia de métodos, de la que AdaBoost es la variante más conocida. Estos métodos trabajan asignando un peso a cada ejemplo. Inicialmente, todos los ejemplos tienen el mismo peso. En cada iteración, se construye un clasificador, denominando base o débil, utilizando algún método de aprendizaje, y teniendo en cuenta la distribución de pesos. A continuación, el peso de cada ejemplo se reajusta, en función de si el clasificador base le asigna la clase correcta o no. El resultado final se obtiene mediante voto ponderado de los clasificadores bases.

Algoritmo Adaboost:

1. Empezar con pesos  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$   $F(x) = 0$
2. Repita el procedimiento para  $m = 1, 2, \dots, M$ :
  - (a) Ajustar la función de regresión  $f_m(x)$  con mínimos cuadrados ponderados de  $y_i$  hacia  $x_i$  con pesos  $w_i$
  - (b) Actualizar  $F(x) \leftarrow F(x) + f_m(x)$
  - (c) Actualizar  $w_i \leftarrow w_i e^{-y_i f_m(x_i)}$  y normalizar
3. Clasificador de salida  $\text{sign}[F(x)] = \text{sign}[\sum_{m=1}^M f_m(x)]$

## Maquina de soporte vectorial

Las *máquinas de soporte vectorial* o *support vector machines* (SVM) surgieron como un método de clasificación basado en la teoría de minimización de riesgo estructural de Vapnik (Vapnik, 1995). En la actualidad tiene numerosas aplicaciones debido a su versatilidad y a sus prestaciones. Las SVM se han utilizado con éxito en campos como la recuperación de información, la categorización de textos, el reconocimiento de escritura

o la clasificación de imágenes, por mencionar algunos; siendo utilizado entre las comunidades dedicadas al aprendizaje automático. Tomando como entrada dos conjuntos de muestras<sup>4</sup> en un espacio de  $n$  dimensiones, el objetivo de este algoritmo es encontrar el hiperplano óptimo (frontera) que maximiza el margen entre los dos conjuntos de datos. Para ello se extraen los ejemplos más cercanos a la frontera, que se conocen como *vectores de soporte* o *support vectors*. El hiperplano óptimo es aquel que maximiza el margen o distancia entre la frontera y dichos vectores de soporte. En general, cuanto mayor sea el margen de separación menor será el error de generalización del clasificador.

Cada ejemplo se representa como un vector de dimensión  $n$  (una lista de  $n$  números), siendo el objetivo separar dichos ejemplos con un hiperplano de dimensión  $n - 1$ , Es lo que se conoce como *clasificador lineal*. Más formalmente, el entrenamiento se representa como un conjunto de pares instancias-clase  $(x_i, y_i)$  tal que  $i = 1 \dots m$ , siendo  $m$  el número de muestras,  $x_i \in R_n$  el vector de características y  $y_i \in \{1, -1_m\}$  la etiqueta que indica si la muestra  $x_i$  pertenece o no a la clase  $y_i$ . SVM obtiene la solución al siguiente problema de optimización:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (9)$$

siendo

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \quad (10)$$

$$\xi_i \geq 0 \quad (11)$$

donde la función  $\langle w, x_i \rangle + b$  representa el hiperplano buscado;  $C$  es un parámetro de compromiso entre el error cometido  $\geq_i$  y el margen; finalmente  $w$  es un vector de

---

<sup>4</sup>El algoritmo básico de SVM permite discriminar entre muestras pertenecientes a dos clases posibles. Es lo que se conoce como clasificador binario

pesos<sup>5</sup>. Las variables  $\xi_i$  fueron introducidas para abordar problemas que no fueran linealmente separables, permitiendo cierto error de clasificación.

SVM fue diseñada originalmente para resolver problemas de clasificación binaria. Para abordar el problema de la clasificación en  $k$  clases (como es el caso de reconocimiento de objetos multiclase) hay que transformar el problema de clasificación multiclase en múltiples problemas de clasificación binaria (Allwein *et al.*, 2001). Hay dos aproximaciones básicas en este sentido: *uno contra todos* (*one-against-all*), donde se entrenan  $k$  clasificadores y cada uno separa una clase del resto, y *uno contra uno* (*one-against-one*), donde se han de entrenar  $\frac{k(k-1)}{2}$  clasificadores y cada uno discrimina entre dos de las clases. Es importante notar que la estrategia *uno contra uno*, al trabajar con menos muestras, tiene mayor libertad para encontrar una frontera que separe ambas clases. Respecto al costo de entrenamiento, es preferible el uso de *uno contra todos* puesto que sólo hay que entrenar  $k$  clasificadores.

Pese a que en su forma más básica SVM induce separadores lineales, si el conjunto no es linealmente separable puede extenderse el algoritmo mediante una transformación no lineal  $\varnothing(X)$  a un nuevo espacio de características. La función  $\varnothing$  permite transformar el espacio de características de entrada (*input space*) en un espacio de trabajo de mayor dimensionalidad (*transformed features space*) donde se puede intentar encontrar de nuevo el hiperplano óptimo. De esta forma se realiza una clasificación lineal con el nuevo espacio, que es equivalente a una clasificación no-lineal en el espacio original.

Las *funciones núcleo* (*Kernel functions*, *funciones kernel* o simplemente *kernels*)

---

<sup>5</sup>El producto escalar de dos vectores  $\mathbf{a} = \{a_1, a_2, \dots, a_n\}$  y  $\mathbf{b} = \{b_1, b_2, \dots, b_n\}$  se define como  $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$

son un tipo especial de función que permite hacer la transformación del espacio de características de forma implícita durante el entrenamiento, sin necesidad de calcular explícitamente la función  $\phi$  (Schölkopf y Smola , 2001; Shawe-Taylor y Cristianini , 2004). Es lo que se conoce como *kernels trick*. Una vez que el hiperplano se ha creado, la función kernel se emplea para transformar los nuevos ejemplos al espacio de características para la clasificación.

Cuatro kernels habituales en tareas de clasificación.  $\gamma$ ,  $r$  y  $d$  son los parámetros de los kernels:

**Lineal**  $k(x_i, x_j) = \langle x_i, x_j \rangle$

**Polinómico**  $k(x_i, x_j) = (\gamma \langle x_i, x_j \rangle + r)^d, \gamma > 0$

**RBF (funciones de base radial)**  $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$

**Sigmoide**  $k(x_i, x_j) = \tanh(\gamma \langle x_i, x_j \rangle + r)$

Formalmente, un kernel  $k$  es una función simétrica  $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = k(x_j, x_i)$  que puede ser interpretada como una medida de similitud entre dos vectores de características  $x_i$  y  $x_j$ . Cabe destacar que la elección del kernel apropiado es importante ya que es éste el que define el espacio de trabajo transformado donde se llevará a cabo el entrenamiento y la clasificación. Existen diversas implementaciones disponibles para la comunidad científica entre las que podemos destacar SvmFU<sup>6</sup>, LIBSVM<sup>7</sup> (Chang y Lin , 2001), SVMLight<sup>8</sup> y Weka<sup>9</sup> (referencia)

---

<sup>6</sup><http://five-percent-nation.mit.edu/SvmFu/>

<sup>7</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

<sup>8</sup><http://svmlight.joachims.org/>

<sup>9</sup><http://www.cs.iastate.edu/~yasser/wlsvm/>

Para poder clasificar con las máquinas de soporte vectorial, se comienza realizando una etapa de aprendizaje. Esta consiste en encontrar el hiperplano  $h(x) = 0$  que mejor separe un conjunto de datos  $X \in \mathbb{R}^d$  según la clase  $Y \in \{-1, 1\}$ , a la que pertenecen. Dicho hiperplano (vease la Figura 47) se corresponde con el que maximiza la distancia al punto más próximo de cada clase, por lo tanto, estará a la misma distancia de los ejemplos más cercanos a cada categoría.

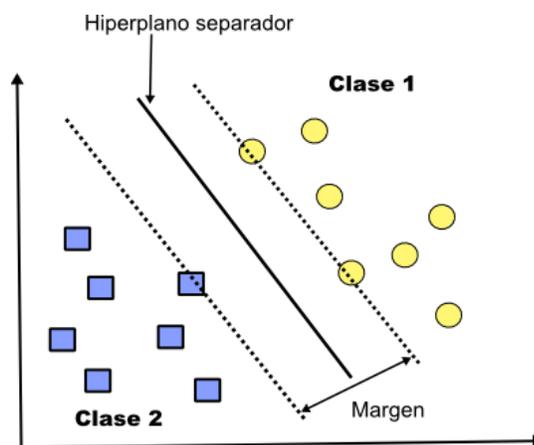


Figura 47: Separación de un conjunto de datos mediante SVM

Según la teoría de Vapnik, el separador lineal que maximiza el margen (2 veces la distancia al punto más próximo de cada clase) es el que nos da la mayor capacidad de generalización, es decir, la capacidad de distinguir características comunes de los datos de cada clase que permitan clasificar imágenes que no sean las del conjunto de entrenamiento. Para ello, es necesario resolver un problema de optimización usando técnicas de programación cuadrática. A los datos que se utilizan para hallar la frontera de decisión (el hiperplano), se les conoce como *vectores de entrenamiento o de aprendizaje*.

A partir de unos datos de entrada  $x_i$ , las SVM nos proporcionarán su clase según la regla de clasificación  $f(x_i) = \text{signo}(h(x_i))$ .

Tras la fase de aprendizaje, se comprueba el error cometido tomando otra muestra de datos (*denominado conjunto de test o validación*) y comparando la salida que obtenemos con su clase real. De una muestra de datos se suele tomar habitualmente un 75% como vectores de aprendizaje y el 25% restante se utiliza para poder comprobar la fiabilidad de la frontera de decisión obtenida.

## V.2.2 Clasificación no supervisada

Un algoritmo de agrupamiento (en inglés, *clustering*) es un procedimiento de agrupación de una serie de vectores cuyos criterios habitualmente de distancia. Así, se trata de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características más comunes (Duda *et al.*, 2001).

Un clasificador de clustering representa un problema ya que los datos pueden revelar clusters de diferentes formas y tamaños. Además, el número de clusters en los datos dependen de la resolución con que vemos los datos. Este proceso de clasificación consiste en: Agrupar un conjunto de  $n$  objetos, definidos por  $p$  variables, en  $c$  clases, donde en cada clase los elementos posean características afines y sean más similares entre sí con respecto a los elementos pertenecientes a otras clases.

La similaridad entre las observaciones se establece en términos de distancia tal como se expondrá en esta sección. El número,  $c$ , de clases puede estar preestablecido o no, y depende del método elegido.

Varios son los propósitos que pueden conducir a este tipo de clasificación:

- Graficar grupos afines, como es el caso de los dendogramas de las taxonomías.
- Clasificar, simplemente, información abundante y compleja.
- Hallar el número  $c$  de clases adecuadas.
- Encontrar subclases dentro de clases naturales.
- Conceptualizar, es decir, interpretar los patrones analizando las causas intrínsecas de la formación de los mismos.
- Hallar clases ocultas no previstas.
- Preprocesar datos complejos con la finalidad de reducir la información a la aportada por los centros de las clases, para posteriormente realizar otros análisis con esta información simplificada

Los métodos que se consideran dentro de esta clasificación son:

- Análisis de conglomerados:
  - Directos
  - Jerárquicos
- Mapa de Kohonen
- Mapa de Ultsch

## Capítulo VI

# Reconocimiento de objetos inspirado en la corteza visual

La mayoría de los sistemas de uso general usan características basadas en niveles de intensidad de la imagen (Agarwal y Roth , 2002; Fei-Fei *et al.* , 2004; Fergus *et al.* , 2003), lo cual se ha mostrado inadecuado debido a los cambios de iluminación. En general, tampoco son adecuados las aproximaciones basadas en características frecuenciales de la imagen. Como ejemplo, las transformadas de wavelet (Schneiderman y Kanade , 2000; Viola y Jones , 2001). En otras palabras, los sistemas implementados hasta el momento están diseñados para clases específicas de objetos. Por ejemplo, los sistemas desarrollados para detectar rostros, detectar automoviles o reconocer ciertas texturas. Sin embargo no se enfocan en el problema de reconocer todo tipo de objetos como lo haría un niño al reconocer rápidamente diferentes clases de objetos. Bajo esta idea surge la necesidad de reconocer clases de objetos siguiendo un mismo criterio, en nuestro caso siguiendo el modelo biológico de los humanos. Bajo esta idea existe una línea de investigación basada en la neurofisiología que a partir de los trabajos de Hartline sobre los campos receptivos (Hartline , 1940) y continuando con las investigaciones de Hubel y Wiesel acerca de como las neuronas son capaces de responder a niveles cada vez más profundos en la trayectoria visual (Hubel y Wiesel , 1962; Hubel y Wiesel , 1968) combinando células simples y complejas en la corteza visual (Hubel , 1988). De esta forma, surgen modelos que ayudan a explicar como esas células pueden transmitir

información cada vez más compleja al grado de poder de reconocer ciertas formas (Hietanen y Perrett , 1993). Bajo toda esa teoría empiezan a surgir propuestas desde los años 80's que intentan representar un modelo computacional inspirado en estas ideas. El primer modelo jerárquico diseñado para el reconocimiento de objetos invariante a ciertas translaciones, llamado Neocognitron (Fukushima , 1980; Fukushima , 1982) es uno de los modelos de red neuronal más complejo que existe, basado principalmente en la estructura del nervio óptico humano, adoptando una arquitectura por capas. Existen otros trabajos basados en redes neuronales multicapa basados en la técnica del gradiente por medio de la convolución (LeCun *et al.* , 1998)

Hoy en día el modelo que más se aproxima al funcionamiento de la corteza visual humana, es el de Riesenhuber y Poggio (Riesenhuber y Poggio , 1999), describiendo un modelo jerárquico enfocado al reconocimiento de objetos inspirado en la ruta ventral; este inicia en la corteza visual primaria (V1), continua en otras áreas visuales como: V2 y V4, llegando a la corteza Infero Temporal, terminando en la corteza Pre Frontal (véase Figuras 3), éste modelo se encuentra basado en la operación de máximos (MAX) de las células simples y complejas emulando la entrada de información a través de las áreas corticales 7). Este modelo le permitió a Serre (Serre *et al.* , 2004; Serre *et al.* , 2005b; Serre *et al.* , 2005a; Serre *et al.* , 2007b), desarrollar una técnica de categorización que, basada en características perceptuales, presenta propiedades interesantes con un bajo número de muestras para su entrenamiento y una gran capacidad de generalización para un reconocimiento de objetos robusto (Serre *et al.* , 2007a). Este modelo combina tolerancia a la posición y a la escala en base a detectores de bordes de varios tamaños.

Otro trabajo importante es el de Mutch y Lowe basados en el trabajo de Serre y

Poggio (Serre *et al.* , 2007a), donde incorporan y mejoran en varias formas el modelo, motivado por propiedades biológicas plausibles en los niveles de las capas de la jerarquía. De esta forma, restringen el número de características de entrada, así como la selección de las mismas, incrementando la esparcificación, inhibición y localización de características conforme avanza el modelo (Mutch y Lowe , 2006; Mutch y Lowe , 2008). Además, existen otros enfoques a través de un método no supervisado en la obtención de las características (Ranzato *et al.* , 2007).

## VI.1 Implementación del modelo HMAX

La aproximación propuesta por Poggio et al. (Riesenhuber y Poggio , 1999; Serre *et al.* , 2004; Serre *et al.* , 2005b; Serre *et al.* , 2005a; Serre *et al.* , 2007b) se basa en el SVH. Dicho sistema está categorizado por la combinación en sucesivas etapas de las respuesta de las células de la corteza visual. El método es denominado HMAX (Hierarchical Model and X) por sus autores (Riesenhuber y Poggio , 1999), o también llamado “Modelo Estándar” (Mutch y Lowe , 2006). Básicamente, dicho sistema consta de 4 capas de unidades de procesamiento (ver Figura 48), donde las unidades **simples S**, se alternan con unidades **complejas C**. Estas características presentan invarianza a translación y escala. No tanto así como la rotación, ya que el modelo, no contempla dicha propiedad. En la Figura 48 presentamos una descripción del método basado en Serre (Serre *et al.* , 2007a).

A continuación se describe con detalle el modelo a utilizar en esta tesis así como su implementación en tres etapas para la **Implementación del Modelo:**

1. Primera Etapa

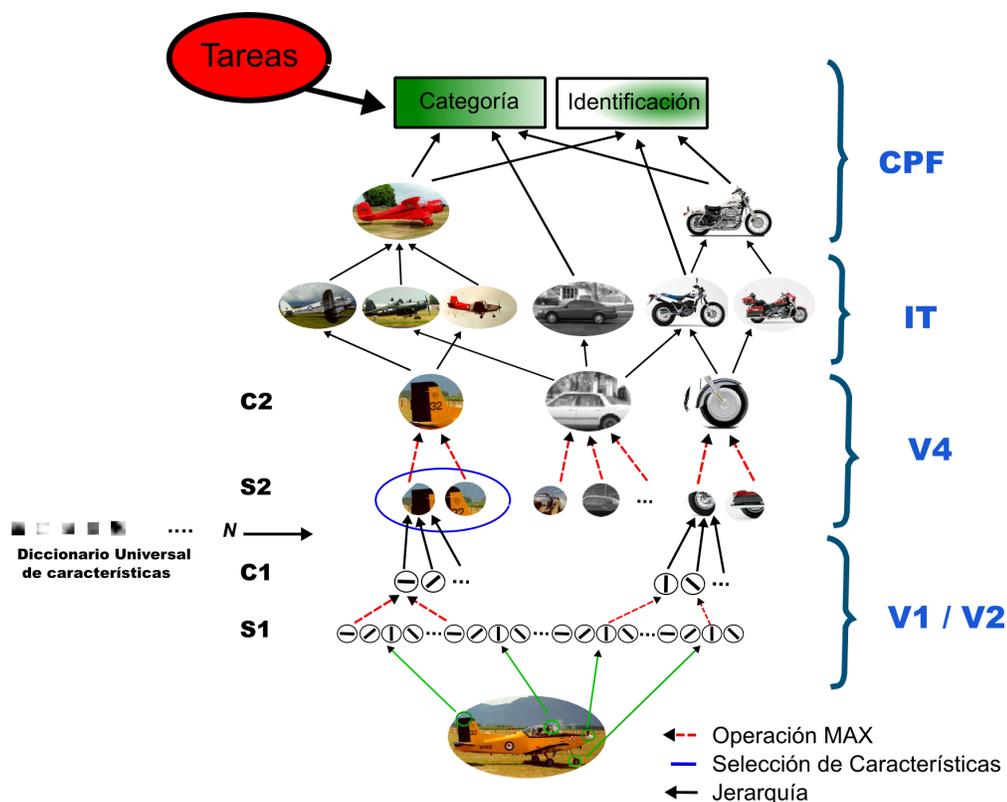


Figura 48: Esquema gráfico del modelo jerárquico. Cada nivel recibe las entradas de un mayor número de unidades llamadas “celulas”. El diccionario de características está compuesto de 200 parches con su cuatro orientaciones de Gabor ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ )

- Extracción de Características (Parches de Imagen)<sup>1</sup> S1 y C1 para genera un diccionario de características.

## 2. Segunda Etapa

- Generar un Vector de Características (S1, C1, S2 Y C2)

## 3. Tercera Etapa

<sup>1</sup>El diccionario universal nos permite resaltar características importantes en la imagen

- Clasificación de Clase de Objetos

### VI.1.1 Capa de filtros de Gabor - S1

Para imitar las características de las unidades (células) correspondientes del modelo se emplean filtros de Gabor, que se ha comprobado coinciden bien con datos fisiológicos de la corteza del gato (Jones y Palmer , 1987). Las características simples S1 se obtienen mediante la aplicación, a una imagen dada un banco de filtros de gabor (Serre *et al.* , 2005b). A continuación se muestra la función para generar los kernel's de Gabor: el filtro de Gabor viene dado por la siguiente expresión:

$$G(x, y) = \exp\left(-\frac{X^2 + \gamma Y^2}{2\sigma^2}\right) x \cos\left(\frac{2\pi}{\lambda} X\right) \quad (12)$$

$$\text{Donde : } X = x \cos\Theta + y \sin\Theta \text{ and } Y = -x \sin\Theta + y \cos\Theta$$

Donde  $\gamma$  es la relación de aspecto horizontal/vertical,  $\Theta$  es la orientación del kernel (ver Figura 49) y  $\sigma$ ,  $\lambda$  son los parámetros relacionados con la amplitud y varianza de la onda (Movellan , 2009).

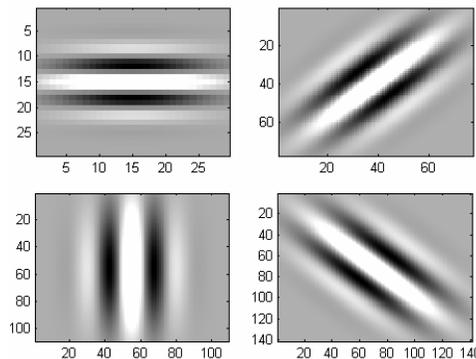


Figura 49: Ejemplo de filtros de Gabor de una escala dada para las 4 orientaciones usadas ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ ).

Serre propone en (Serre *et al.* , 2004) un conjunto de parámetros, experimentales obtenidos, que responden de forma análoga a las células de la ruta ventral. En concreto, se usan 16 escalas con 4 orientaciones por escala. En la Figura 49 podemos ver un ejemplo de los 4 filtros de una determinada escala.

### Detalles de implementación

El modelo es implementado como primera herramienta en Matlab para el procesamiento de imágenes y para la parte de clasificación se utiliza SVM con ayuda de la librería LIBSVM.

Todas las imágenes para los experimentos fueron normalizadas a un tamaño de 140 pixeles de alto preservando el aspecto de radio<sup>2</sup> en el ancho. Antes de aplicar los filtros de gabor se realiza una etapa previa. En ella se genera un piramide de imágenes en escalas diferentes considerando un factor de escala<sup>3</sup>, ver la Figura 50.

Para la implementación de los filtros de Gabor se utilizaron los valores experimentales calculados por Serre y Poggio (Serre *et al.* , 2005b) que simulan las células simples de la corteza visual.

$\theta$  Orientación del filtro ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ ).

$\gamma$  Relación de aspecto (0.3).

$\sigma$  Ancho (4.5).

$\lambda$  Longitud de Onda (5.6)

En la Figura 51 se muestra como se visualiza una imagen aplicando filtros de Gabor

---

<sup>2</sup>Proporción entre el ancho y el alto

<sup>3</sup>Factor de Escala:  $2^{1/4} = 1.1133$



Figura 50: Piramide de imágenes a 12 escalas diferentes.

en sus 4 orientaciones ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ ) y en la Figura 52 se visualizan las piramides de escalas convolucionadas. Es decir por cada escala se genera una banda que contiene 4 mapas o imágenes con filtros de Gabor, formando un total de 48 mapas.

### VI.1.2 Capa de invarianza local - C1

Las características complejas C1 se obtienen a partir de las características S1 mediante los operadores de máximos. Para cada orientación se realizan los pasos:

1. En este paso se obtienen 6 bandas, donde cada banda se obtiene por submuestreo, mediante la aplicación del operador de máximo entre las 2 escalas.
2. Posteriormente, se realiza una operación de máximos locales en la imagen originada por las 2 escalas, ver Figura 54

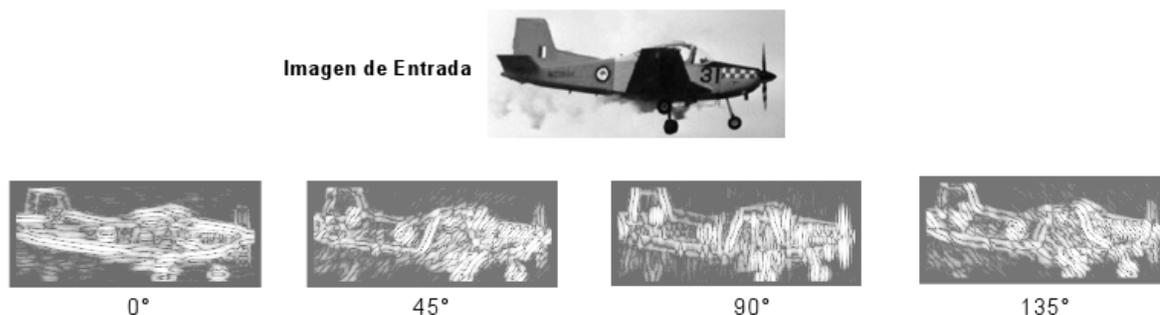


Figura 51: Imagen convolucionada con los filtros de Gabor en sus diferentes orientaciones.

Dentro de esta etapa se hace la extracción de características (parches de imagen) para generar un diccionario de parches, cada parche varían en tamaño de  $3 \times 3$  o  $5 \times 5$ , manteniendo las 4 orientaciones de gabor ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ ), Por lo tanto, cada parche está compuesto de 4 subparches.

### Detalles de Implementación

A partir de las 12 escalas se generan 6 bandas, tomando las escalas 1 y 2 para generar la primera banda y así sucesivamente, reescalando la segunda escala al mismo tamaño que la primera (véase la Figura 53).

En esta etapa se consigue invarianza a traslación (máximos locales por submuestreo) y a escala (máximo entre 2 escalas). En el máximo local se toma una rejilla de tamaño  $8 \times 8$  y saca el máximo, se recorre 5 espacios y se toma nuevamente otra rejilla de tamaño. Por lo tanto existe un traslape de 3 rejillas. Cada banda mantiene las 4 orientaciones. Durante el proceso, todos los máximos se definen por su valor absoluto.

Durante esta etapa se genera un diccionario de características (ver Figura 56), es

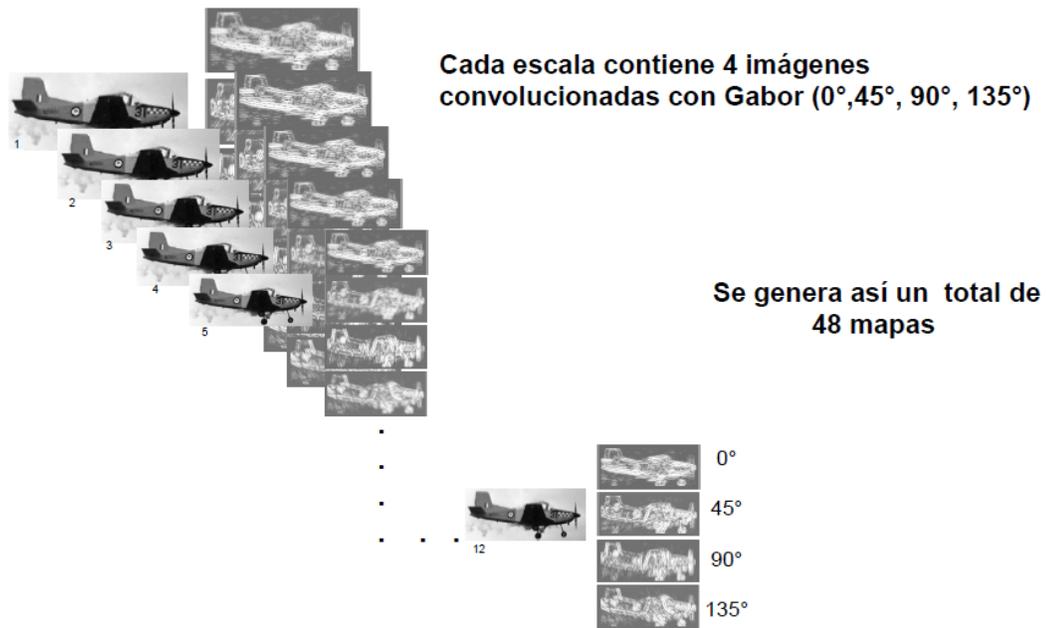


Figura 52: Por cada escala se generan 4 imágenes convolucionadas con filtros de gabor, generando un total de 48 imágenes.

decir, se toma la imagen de entrada y se extraen un parche de la imagen aleatoriamente (ver Figura 57) cumpliendo con un valor de umbral de 0.2. El tamaño del parche de la imagen es de un tamaño de  $3 \times 3$  o de  $5 \times 5$  en cada orientación de Gabor ( $0^\circ, 45^\circ, 90^\circ$  y  $135^\circ$ ).

### VI.1.3 Capa de características locales - S2

Durante la fase entrenamiento, solamente, por cada imagen de entrenamiento, y a partir de cada mapa (Serre *et al.*, 2005b), se extraen  $K$  parches  $P_i$  de diferentes tamaños  $n_i \times n_i$  con sus 4 orientaciones (cada parche es de tamaño  $n_i \times n_i \times 4$ ).

Para cada posible parche  $X$  en cada posición del mapa ( $\cdot$ ), se calcula:

$$Y = \exp(-\gamma \|X - P_i\|^2) \quad (13)$$

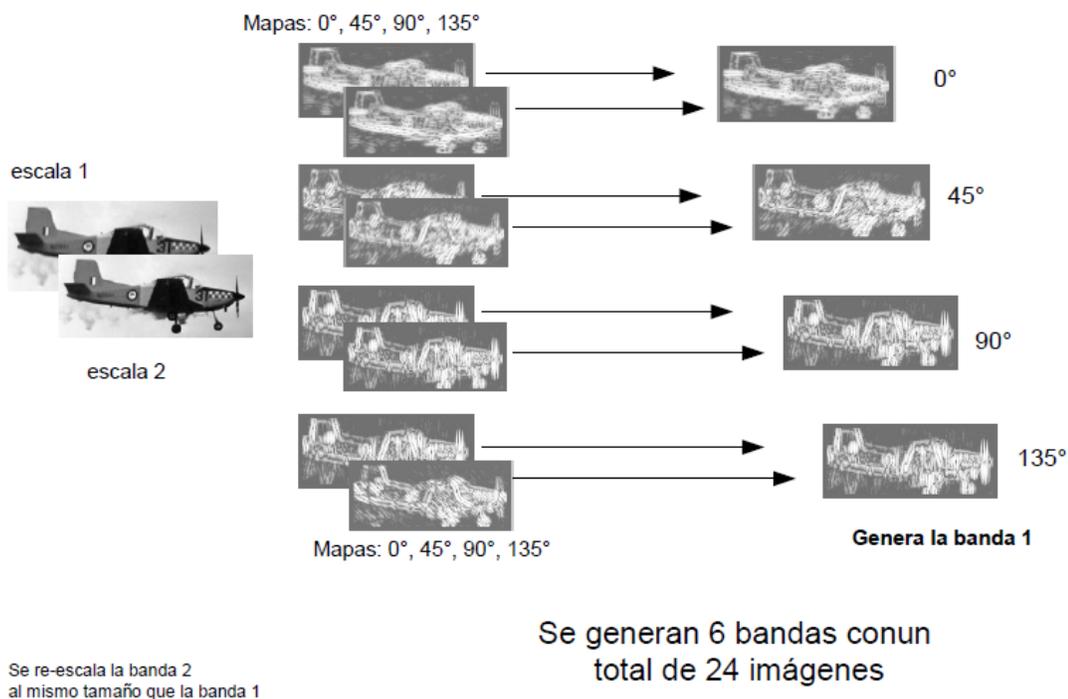


Figura 53: En la Operación Max el número 1 se realiza primero entre las escalas

Donde  $\gamma$  es un parámetro que controla el grado de similitud de los parches comparados. Esto se hace para cada banda y para cada parche  $P_i$  de forma independiente. De este modo obtenemos el mapa, donde sus valores se mueven entre  $[0,1]$ .

### Detalles de implementación

Se toma la imagen de entrada, se calcula la etapa C1, ver Figura 55; es decir se obtienen 4 mapas, se elevan al cuadrado por separado y se suman los 4 mapas generando uno solo, este se multiplica por 1's y se le suma el parche del diccionario seleccionado, además, el resultado anterior se convoluciona con el parche de cada orientación obtenido en el diccionario de parches. El resultado de cada convolución lo multiplica por 2 y se le resta a la imagen obtenida. El resultado de estas operaciones se le aplica la RBF para

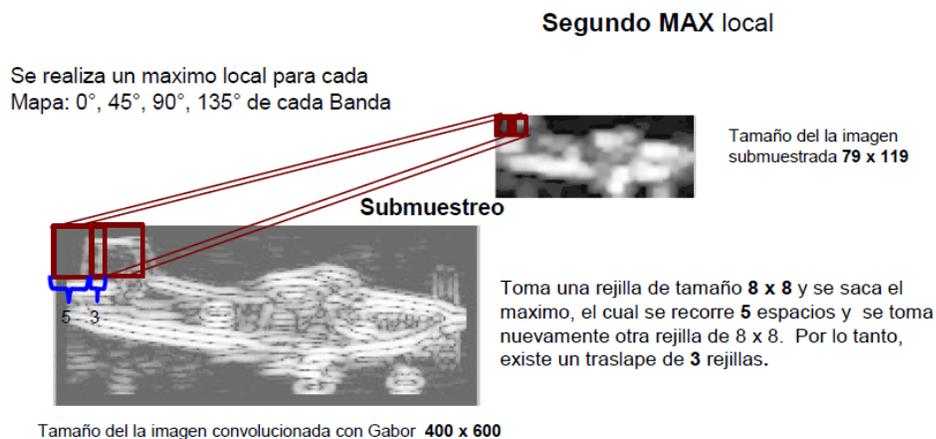


Figura 54: En la Operación MAX el número 2 se generan a partir de la imagen entre 2 escalas generando un total de 6 bandas con un total de 24 imágenes

obtener el mapa S2. Estas operaciones se realizan en las 4 orientaciones de gabor

#### VI.1.4 Capa de invarianza global - C2

En este punto, para cada mapa, calculamos el máximo global entre todas las posiciones, y, posteriormente, el máximo global entre las 6 bandas. De este modo, obtenemos la características por cada parche  $P_i$ . Aquí se consigue una nueva invarianza a escala y translación.

##### Detalles de implementación

Detalles de implementación, se toman los 6 mapas generados en la capa anterior (S2) relacionado a un parche del diccionario de parches y se calcula una operación de máximos de los 6 mapas, generando un vector de características del mismo tamaño del diccionario de parches (100 ó 200), ver Figura 59.

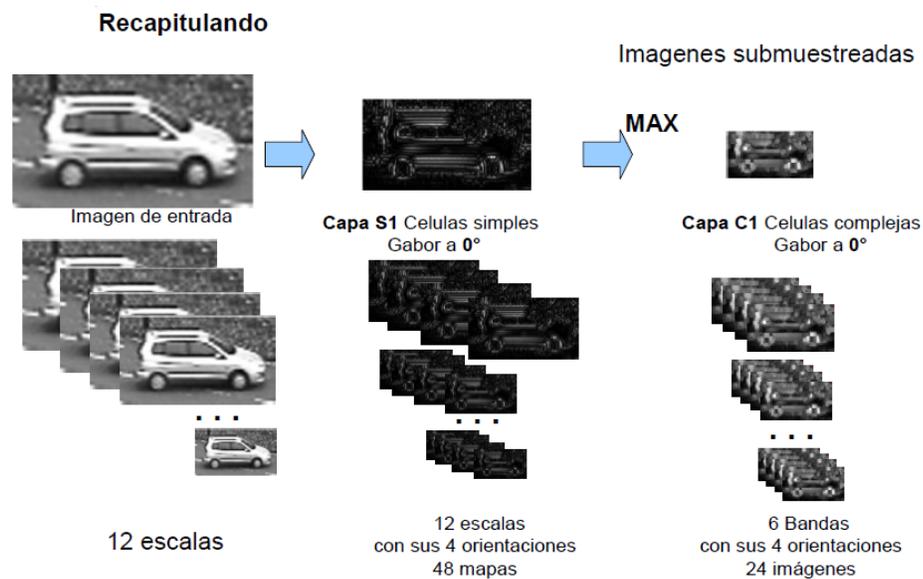


Figura 55: Recapitulando la capa C1.

### VI.1.5 Experimentos

En esta sección se muestran los experimentos realizados y los resultados obtenidos, se encuentran divididos en tres secciones:

1. Se extrae un diccionario de características de las clases de objetos que se están utilizando en esta tesis. El diccionario contiene sólo 100 características.
2. Se utiliza el diccionario universal propuesto por Serre con 200 características.
3. Se realizó una nueva implementación del modelo estándar de la ruta ventral en la tecnología CUDA.

Para realizar las pruebas se utilizaron las bases de datos de Caltech 101, de las cuales se seleccionaron 10 categorías, ver Figura 60. Cada clase contiene 100 imágenes. Las clases seleccionadas son:

1. Aviones
2. Bonsais

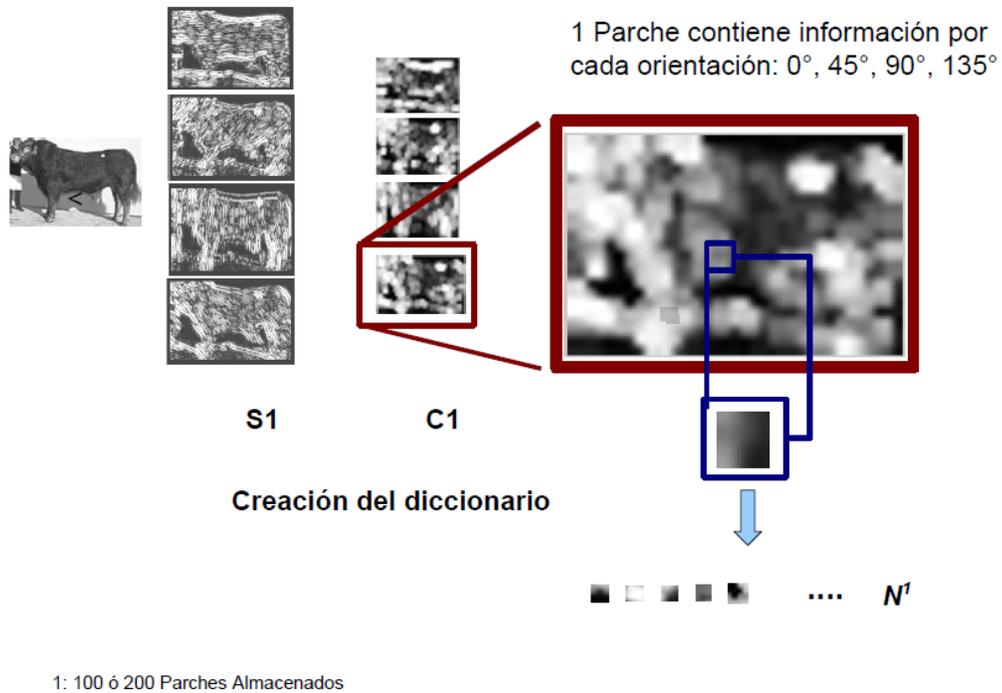


Figura 56: Selección de los parches.

3. Cerebros
4. Carros
5. Sillas
6. Rostros
7. Hojas
8. Motos
9. Goletas<sup>4</sup>
10. señal de Stop

Todos los experimentos se hicieron en una PC Dell Precision T7500 Workstation, Intel Xeon, con 8 Core, 4 GB en RAM y una tarjeta gráfica Quadro FX 3800, con sistema operativo Linux OpenSUSE 11.1.

Para cada corrida se siguen los siguientes pasos:

<sup>4</sup>Goleta: es un buque de dos o más mástiles.

## Creación del diccionario

Se extraen **N** parches o prototipo, (100 y 200) son suficientes, para reconocer objetos.

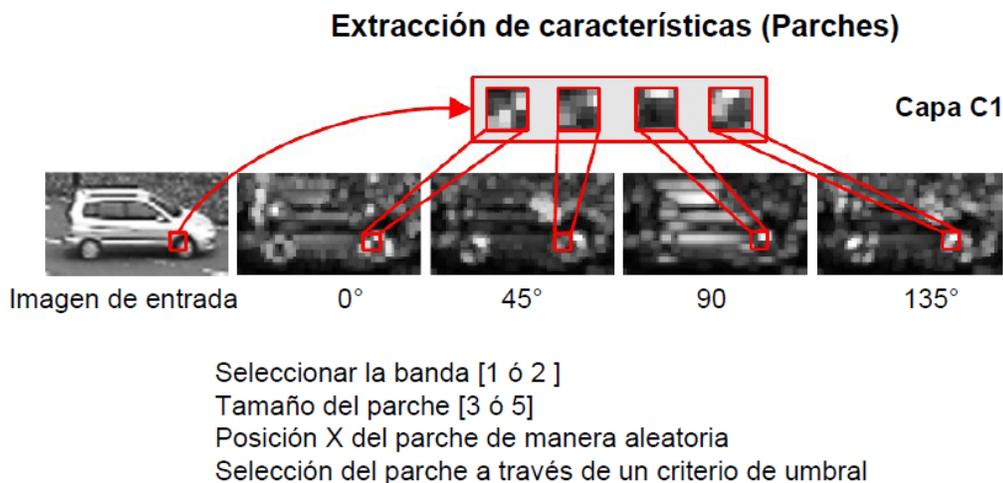


Figura 57: Creación del diccionarios de características (features)

1. Se seleccionaron 15 ó 30 imágenes de entrenamiento aleatorias de cada categoría, el resto de las imágenes se utilizan para la prueba.
2. Construir un vector de características para el conjunto de entrenamiento
3. Construir un nuevo vector de características para el conjunto de imágenes de prueba.
4. El entrenamiento y la etapa de prueba se realizan utilizando SVM<sup>5</sup>.

## Experimento 1

Para realizar los experimentos, todas las imágenes para los experimentos fueron normalizadas a un tamaño de 140 pixeles de alto preservando el aspecto de radio<sup>6</sup> en

<sup>5</sup>LIBSVM, disponible en <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

<sup>6</sup>Proporción entre el ancho y el alto

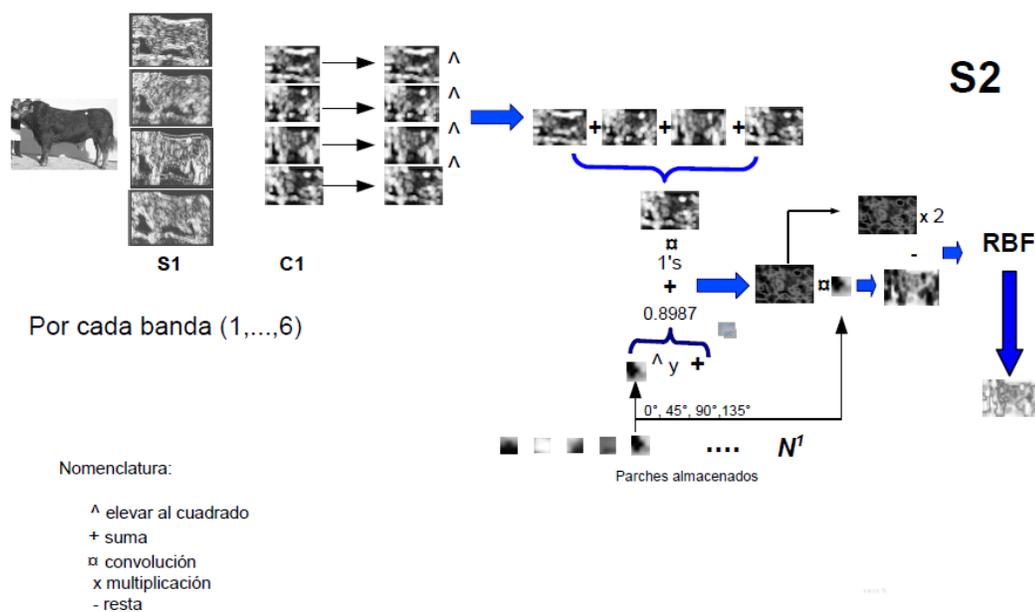


Figura 58: Operaciones que se realiza por cada parche en las Capas S2.

el ancho. Antes de aplicar los filtros de gabor se realiza una etapa previa. En ella se genera un piramide de imágenes en escalas diferentes considerando un factor de escala<sup>7</sup>.

Se crea un diccionario de características de las categorías seleccionadas, ver Figura 61, con solo 100 características<sup>8</sup>

En el experimento número uno se obtuvieron los siguientes resultados. Éste resultado es el promedio de 8 corridas; con un entrenamiento de el 92% de clasificación y en la etapa de prueba un 76% de clasificación. Se muestra un ejemplo de las imágenes clasificadas en la Figura 62 en forma de tabla.

<sup>7</sup>Factor de Escala:  $2^{1/4} = 1.1133$

<sup>8</sup>Cada características está compuesto de 4 parches en orientaciones de (0°, 45°, 90°y 135°)

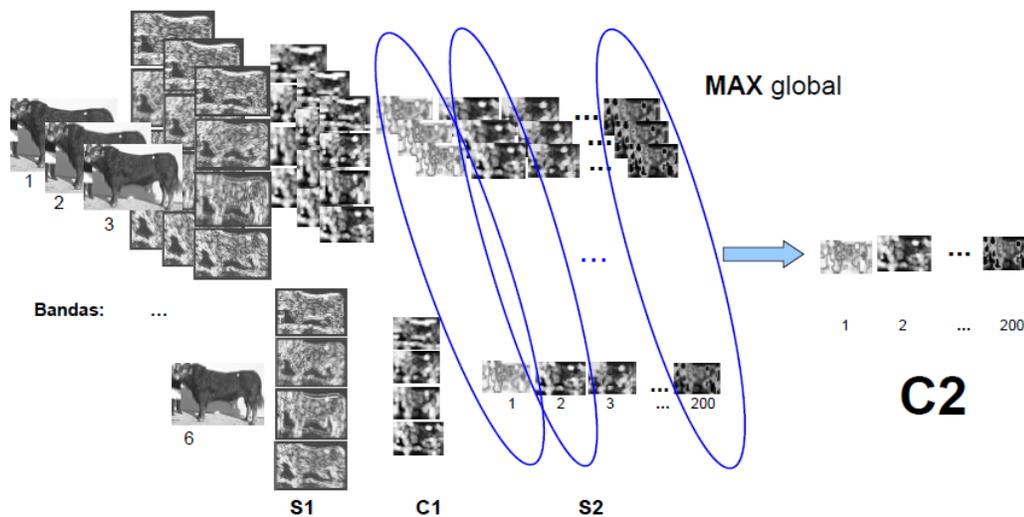


Figura 59: Operacion de MAX

## Experimento 2

En este experimento se toma el diccionario universal de características propuesto por Poggio (Serre *et al.*, 2005b), éste contiene 200 parches en sus 4 orientaciones de Gabor ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ ). Se realizaron los experimentos en las mismas circunstancias que el experimento 1, ejecutando 8 corridas. Los resultados obtenidos son del 94% de clasificación en el entrenamiento y 73% de clasificación en la prueba. Se muestra un ejemplo de las imágenes clasificadas en la Figura 63 en forma de tabla.

## Experimento 3

Se desarrolló una implementación del modelo en la tecnología GPGPU o “*General-Purpose Computing on Graphics Processing Units*”.

Como se menciona en el capítulo IV, la GPU es un procesador diseñado para los cálculos implicados en la generación de gráficos. Algunas de sus características son: el bajo precio en relación a su potencia de cálculo, gran paralelismo, optimización para cálculos en coma flotante. Las cuales se consideran atractivas para su uso en aplicaciones fuera de los gráficos por computadora, especialmente en el ámbito científico y de simulación. Así, se han desarrollado técnicas para la implementación de simulaciones de fluidos, bases de datos, algoritmos de clustering, entre algunos otros.

Para la implementación del modelo se utilizó CUDA (Compute Unified Device Architecture) que hace referencia tanto a un compilador como a un conjunto de herramientas de desarrollo creadas por nVidia que permiten a los programadores usar una variación del lenguaje de programación C para codificar algoritmos en GPUs de nVidia. Por otro lado, por medio de wrappers<sup>9</sup> se puede usar Python, Fortran y Java en vez de C/C++ y en el futuro también se añadirá OpenCL y Direct3D. CUDA funciona en todas las GPUs nVidia de la serie G8X en adelante, incluyendo GeForce, Quadro y la línea Tesla.

Todos los experimentos se hicieron en una PC Dell Precision T7500 Workstation, Intel Xeon, con 8 Core, 4 GB en RAM y una tarjeta gráfica Quadro FX 3800, con sistema operativo Linux OpenSUSE 11.1. Y los drivers de Nvidia CUDA con la versión NVIDIA 190.53.

A continuación se ejemplifica una prueba del modelo estandar de la ruta ventral, comparandolo con Matlab y CUDA, vease la Figura 64.

---

<sup>9</sup>Un Wrapper es un programa que controla el acceso a un segundo programa en otro lenguaje de programación

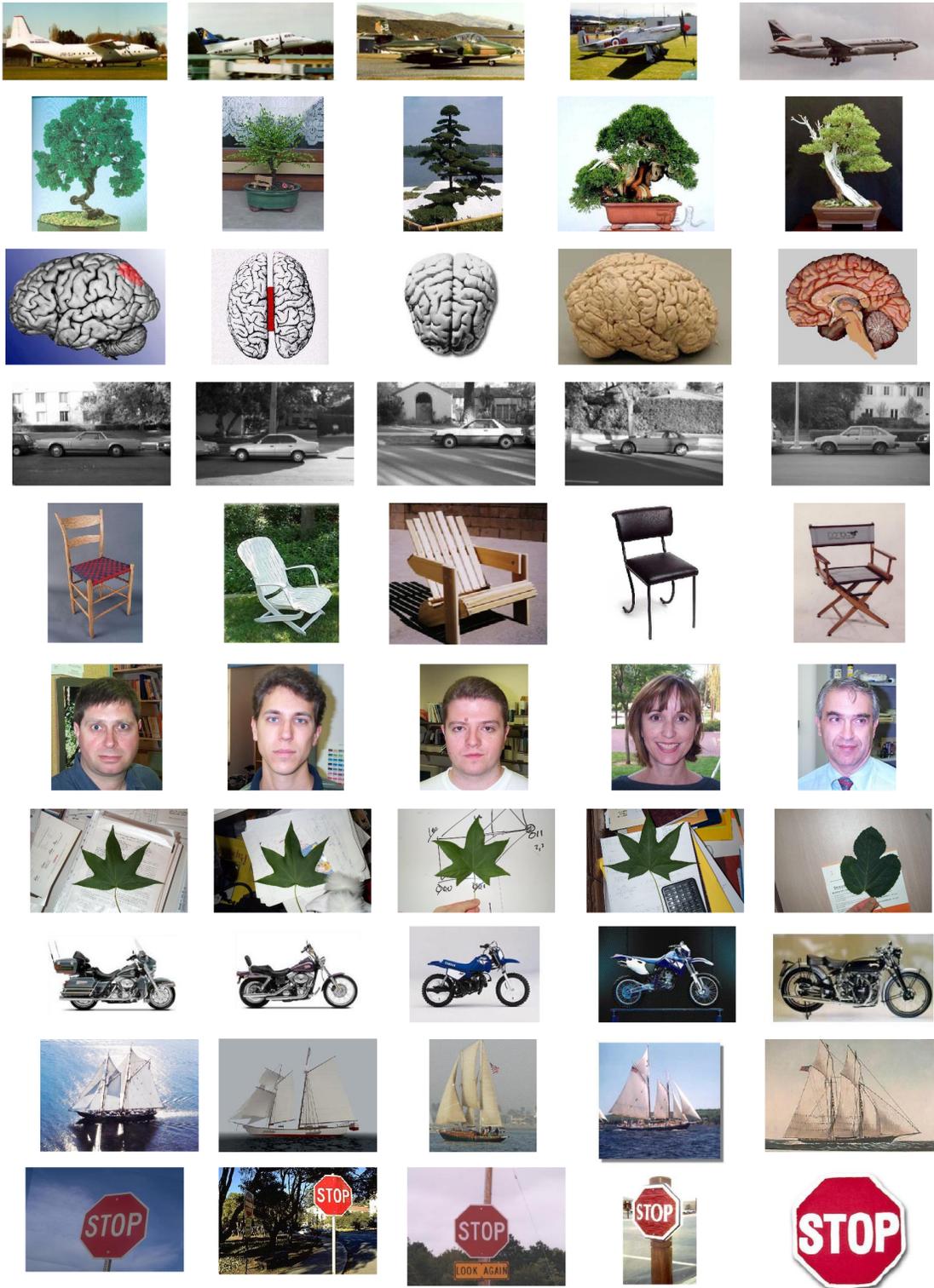


Figura 60: Operacion de MAX

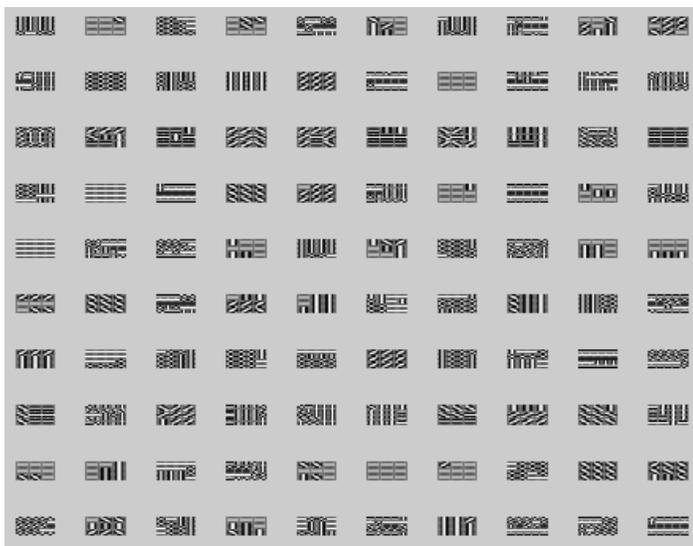


Figura 61: Selección de las características de las clases de objetos

	Aviones	Hojas	Motocicletas	Bonsais	Cerebros	Carros	Sillas	Rostros	Goletas	Señal de STOP
Aviones	<b>12</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Hojas	<b>0</b>	<b>12</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
Motocicletas	<b>1</b>	<b>0</b>	<b>13</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Bonsais	<b>0</b>	<b>0</b>	<b>0</b>	<b>10</b>	<b>3</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
Cerebros	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>12</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>
Carros	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>11</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
Sillas	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>11</b>	<b>0</b>	<b>1</b>	<b>0</b>
Rostros	<b>0</b>	<b>1</b>	<b>0</b>	<b>2</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>10</b>	<b>0</b>	<b>1</b>
Goletas	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>12</b>	<b>0</b>
Señal de STOP	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>11</b>

Figura 62: Matriz de Confusión 76% de clasificación en imágenes de pruebas. Se clasificaron correctamente 114 de 150 imágenes

	Aviones	Hojas	Motocicletas	Bonsais	Cerebros	Carros	Sillas	Rostros	Goletas	Señal de STOP
Aviones	<b>11</b>	<b>0</b>	<b>2</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Hojas	<b>0</b>	<b>12</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
Motocicletas	<b>2</b>	<b>0</b>	<b>11</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Bonsais	<b>0</b>	<b>0</b>	<b>0</b>	<b>10</b>	<b>3</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
Cerebros	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>10</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>
Carros	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>11</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
Sillas	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>11</b>	<b>0</b>	<b>1</b>	<b>0</b>
Rostros	<b>0</b>	<b>1</b>	<b>0</b>	<b>2</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>10</b>	<b>0</b>	<b>1</b>
Goletas	<b>0</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>10</b>	<b>0</b>
Señal de STOP	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>11</b>

Figura 63: Matriz de Confusión 71.33% de clasificación en imágenes de pruebas. Se clasificaron correctamente 107 de 150 imágenes

	<b>Modelo HMAX<sup>1</sup></b>		
	<b>MATLAB</b>	<b>CUDA</b>	
Imagen 896 x 592	34 seg	3.5 seg	
Imagen 601 x 401	24 seg	2.7 seg	
Imagen 180 x 113	9 seg	1 seg.	

1: Extracción de Características y Generación del Vector de Características  
(Etapa Dos y Tres)

Figura 64: Tiempo que toma el modelo HMAX en imágenes de diferentes tamaños.

## Capítulo VII

# Modelo HMAX con Programación Genética

Hoy en día, el uso de Cómputo evolutivo nos ayuda a resolver problemas de visión; lo cuál nos permite explorar espacios de búsqueda enormes para encontrar soluciones que pueden estar lejos del espacio que normalmente se analiza con técnicas tradicionales (Cagnoni *et al.* , 2007). Sin embargo, los problemas de visión representan nuevos retos para el área de aprendizaje de máquina. (Hartley y Zisserman , 2003).

Uno de los trabajos de investigación del grupo EvoVisión, pioneros en la resolución de problemas de visión por medio de técnicas evolutivas; uno de ellos es el problema de reconocimiento de expresiones faciales en imágenes térmicas, proponiendo un Algoritmo Genético (GA) jerárquico. El cromosoma está compuesto de dos partes: en una se selecciona las regiones faciales y en la otra los rasgos de textura para cada región (Hernández *et al.* , 2007). Utilizando además SVM<sup>1</sup> durante la etapa de clasificación. Otro trabajo del mismo grupo es el problema de clasificación de imágenes en base a su contenido a través de textura (Perez y Olague , 2007).

Otro trabajo importante del mismo grupo es el de Trujillo y Olague (Trujillo y Olague , 2006; Trujillo y Olague , 2008). En él se describe una metodología a través

---

<sup>1</sup>SVM: del inglés Support Vector Machine

de programación genética para sintetizar operadores que son de uso general para la detección robusta y dispersa de puntos de interés, el cuál abre un panorama hacia los operadores de evolución dentro del espacio de búsqueda empleados. Además otro trabajo realizado bajo este mismo paradigma, donde Pérez y Olague, desarrolla una tarea de aprendizaje del operador del descriptor de región invariante a ciertas translaciones bajo programación genética (Pérez y Olague , 2008). La idea es encontrar un conjunto de expresiones matemáticas, que sean igual o mejor que la magnitud del gradiente utilizado en el descriptor de SIFT para la extracción de características locales, enfocando la problemática hacia la correspondencia de características del objeto en una escena, a fin de identificar el objeto en ella.

Cómo se menciona en el capítulo anterior de reconocimiento de objetos inspirado en la corteza visual, los sistemas implementados hasta el momento sobre el reconocimiento de clase de objetos, están diseñados para clases específicas de objetos con un elevado número de imágenes de entrenamiento. Bajo esta idea surge la necesidad de reconocer clases de objetos siguiendo un mismo criterio, es por eso que en esta tesis se inspira en el modelo biológico de los humanos, es decir a través del flujo de información visual que se lleva en la ruta ventral o ruta de ¿Qué?, compuesta de las capas: V1, V2 y V4, terminando su procesamiento en la corteza infero temporal. Por lo tanto, a partir de los trabajos de Hartline sobre los campos receptivos (Hartline , 1940) y continuando con las investigaciones de Hubel y Wiesel acerca de cómo las neuronas son capaces de responder a niveles cada vez más profundos en la trayectoria visual (Hubel y Wiesel , 1962; Hubel y Wiesel , 1968) combinando células simples y complejas en la corteza visual en sus diferentes capas (Hubel , 1988). De esta forma, surgen modelos que ayudan a explicar cómo esas células pueden transmitir información cada vez más compleja al grado de poder de reconocer ciertas formas (Hietanen y Perrett , 1993), bajo estas investigaciones

Poggio et al. (Riesenhuber y Poggio , 1999) diseña un “*modelo estandar*” de la ruta ventral con el nombre de HMAX (Hierarchical Model and X), el cuál es utilizado como base en éste trabajo.

## VII.0.6 Detalles de Implementación

El algoritmo propuesto en este Capítulo se programó utilizando Matlab, con el *toolbox* de programación genética GPLAB<sup>2</sup>

Todos los experimentos se hicieron en una PC Dell Precision T7500 Workstation, Intel Xeon, con 8 Core, 4 GB en RAM y una tarjeta gráfica Quadro FX 3800, con sistema operativo Linux OpenSUSE 11.1.

Las bases de datos (ver Figura 64) que se han utilizado en está tesis para realizar los experimentos son las siguientes:

- Base de datos de las 101 categorías de objetos de Caltech (Caltech101 , 2001) de las cuáles se seleccionaron 9 clases de objetos; Aviones, Motocicletas, Bonsais, Cerebros, Carros, Sillas, Rostros, Goletas, y señales de STOP.
- Base de datos seleccionadas del Caltech (Caltech5 , 2005). De esa base de datos solo se selecciono la clase de objetos: Hojas.

Para nuestro trabajo descrito a lo largo de esa tesis. Se define el conjunto de Funciones y Terminales que se utilizaron para definir el espacio de búsqueda:

$$F = \{+, | + |, -, | - |, *, \div, \sqrt{I_t}, \frac{I_t}{2}, \log_2(I_t), D_x G_\sigma, D_y G_\sigma, G_\sigma\}$$

---

<sup>2</sup><http://gplab.sourceforge.net/index.html>, GPLAB A Genetic Programming Toolbox for MATLAB.

$$T = \{I, D_x(I), D_y(I), D_{xx}(I), D_{yy}(I), D_{xy}(I)\} \quad (14)$$

donde  $I$  es la imagen de entrada, y  $I_t$  puede ser cualquier terminal en  $T$ , o la salida de una de las funciones de  $F$ ;  $D_u$  representan derivadas Gaussianas de la imagen en la dirección  $u$ . donde  $D_u = I * G_{u(\sigma=1)}$ ;  $G_\sigma$  son filtros de suavizado Gaussiano con  $\sigma = 1$  ó  $2$ ;  $D_u G_\sigma$  representan la derivada de la imagen de entrada con un filtro Gaussiano con suavizado  $\sigma$ . En el caso de la función  $\log_2$  se toma la convención de  $\log_2(0) = 0$  para evitar operaciones con cantidades infinitas. Por razones similares, cuando se da el caso en que se aplica la raíz a un valor negativo  $\sqrt{-x}$ , solo se toma la parte real del resultado para evitar operaciones posteriores con número complejos,

En la Figura 65 muestra que la parte de evolución que se lleva a cabo con GP es en la capa S2, donde en vez de utilizar el diccionario de características, se generan operadores matemáticos que resalta de mejor manera información en la imagen.

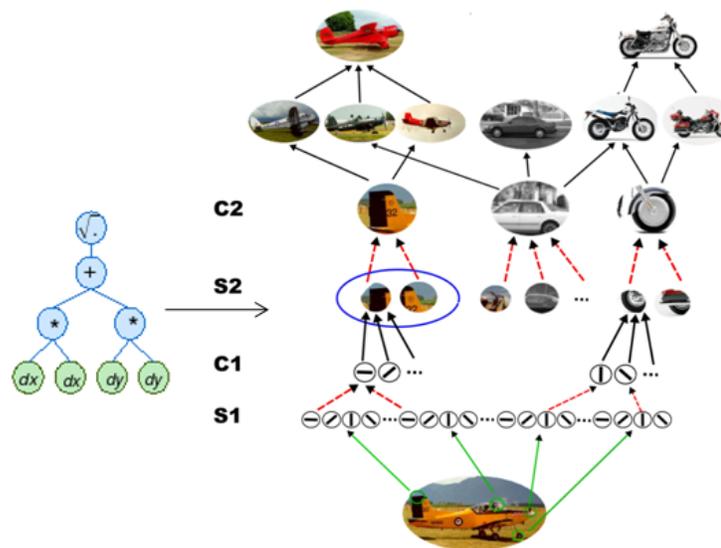


Figura 65: Modelo estándar de la ruta ventral con programación genética.

A continuación se presenta el modelo de evolución con el paradigma de programación genética.

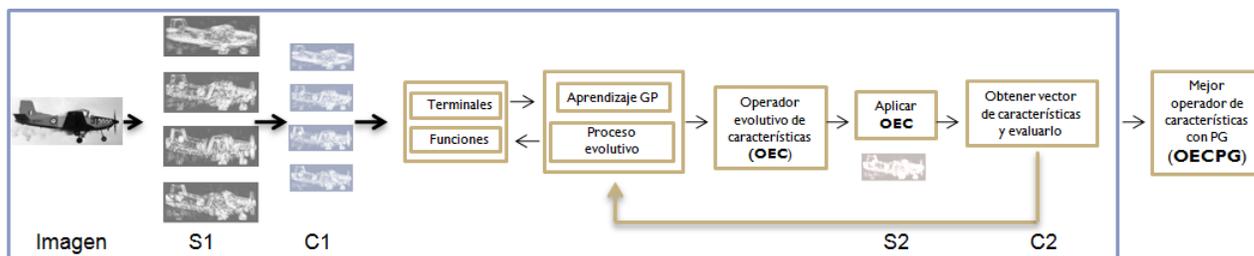


Figura 66: Proceso de evolución de los operadores de características.

Una vez planteado nuestro algoritmo inspirado en la corteza visual a través de programación genética las pruebas se han dividido en 2 secciones. La primera, se refiere a la etapa de entrenamiento, donde se obtuvieron un resultado del 84 % de clasificación, y un 74% de clasificación en la etapa de prueba

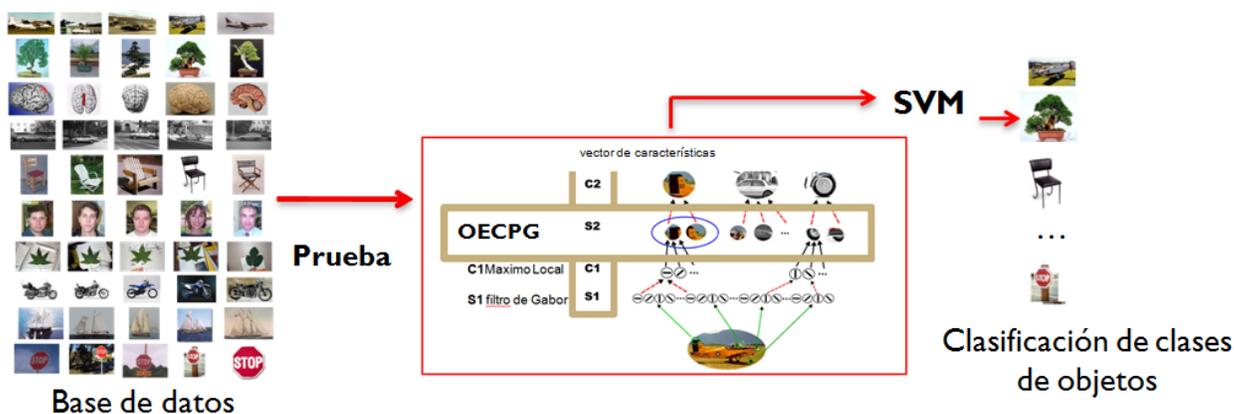


Figura 67: Etapa de evaluación del modelo estár con GP.

En la segunda parte, se genera la etapa de prueba del modelo, el cuál aún se siguen

haciendo la pruebas correspondientes.

En la Tabla I se especifican los parámetros principales del algoritmo GP propuesto.

Tabla I: Parámetros del GP utilizados en el HMAX

Parámetro	Descripción y valor
Tamaño de la población	50 Individuos.
Generaciones	50.
Inicialización	Mitad y Mitad.
Probabilidades de operadores genéticos	Cruce $p_c = 0.90$ ; Mutación $p_\mu = 0.1$
Profundidad de los árboles	Profundidad Dinámica.
Profundidad máxima dinámica	5 niveles
Profundidad máxima real	7 niveles
Selección	Torneo
Supervivencia	Elitismo
Parámetros de la función objetivo	0.7 + 0.3

Los parámetros de la función objetivo son 0.7 de correcta clasificación y 0.3 del individuo con el menor número de árboles.

Los primeros cuatro parámetros, *tamaño de la población*, *generaciones*, *inicialización*, y *probabilidades de operadores genéticos*, tienen valores canónicos, y fueron ligeramente ajustados experimentalmente. Los siguientes tres ayudan a controlar el tamaño de los individuos. El problema conocido como *code bloat* en la literatura de GP. La *profundidad de los árboles* se determina de forma dinámica, para esto se cuenta con con dos valores máximos de profundidad que un árbol puede tener. La *profundidad máxima dinámica* es un límite que los individuos solo pueden sobrepasar si su *aptitud* es igual o mayor a la *aptitud* del mejor individuo obtenido hasta el momento. Cuando esto sucede la *profundidad máxima dinámica* se aumenta al valor de la profundidad de

dicho individuo. El siguiente parámetro, *profundidad máxima real* es un límite rígido que no puede ser sobrepasado en ninguna circunstancia. Finalmente, el manejo de la población se hace con una selección de individuos utilizando *torneo*, y una estrategia *elitista* para la supervivencia de individuos.

La función de aptitud equivale al porcentaje de clasificación de los objetos del conjunto de imágenes dado, utilizando las características extraídas de las imágenes y el porcentaje menor del individuo con el número menor de árboles.

Los vectores de características se codifican de tal manera que la SVM pueda trabajar con ellos. Es decir se extraen de las 6 bandas 200 características importantes.

Una vez que se obtiene la mejor solución del GP; es decir, el individuo que mejor clasifique las imágenes de entrenamiento. En la Figura 68 muestra un conjunto de individuos evolucionados y aplicados a una imagen, obteniendo como resultado una imagen de interés para la extracción de características.

Tabla II: Clases de objetos, se utilizan 30 ó 15 imágenes de forma aleatoria de las 100 imágenes por clase

Clase	Conjunto de entrenamiento	Conjunto de prueba
Aviones	30	30
Bonsai	30	30
Cerebros	30	30
Carros	30	30
Sillas	30	30
Rostros	30	30
Hojas	30	30
Motos	30	30
Goletas	30	30
Señal de stop	30	30

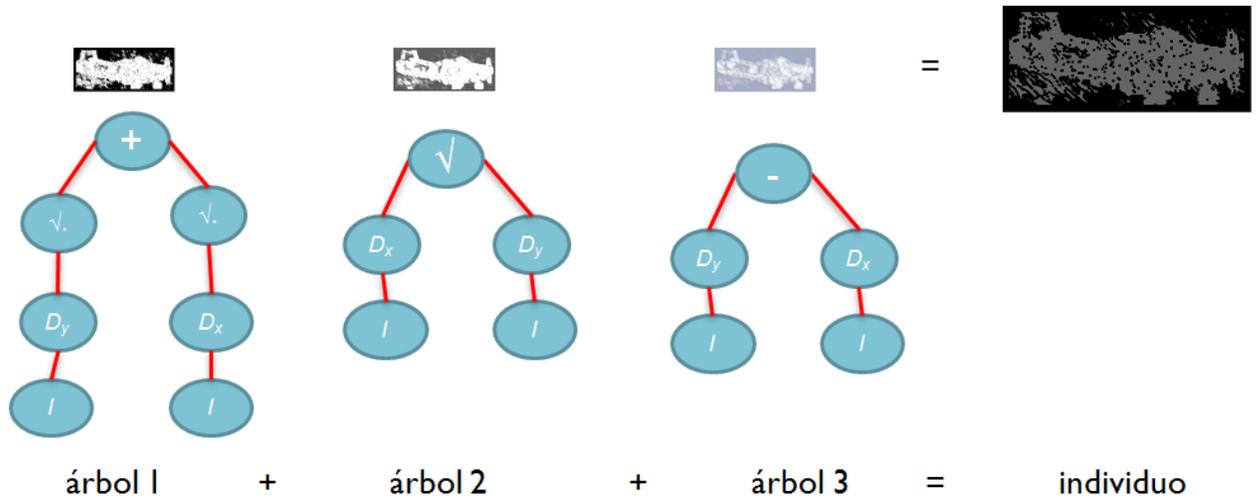


Figura 68: Ejemplo de de operadores matemáticos evolucionados y aplicados a una imagen.

## VII.0.7 Herramientas utilizadas

### GPLAB

*GPLAB* es una caja de herramientas de programación genética para MATLAB, bajo la licencia pública general GNU. Se puede descargar del la siguiente liga <http://gplab.sourceforge.net/>.

Algunas características de GPLAB son:

- 3 modos de inicializar los arboles (Full<sup>3</sup>, Grow<sup>4</sup>, Ramped Half-and-Half<sup>5</sup>).
- Controlar el tamaño de la población.
- 4 operadores genéticos (cruza, mutación, mutación de intercambio).
- 4 niveles de elitismo.

<sup>3</sup>Full: mediante este método los arboles tienen forma regular, es decir, es un árbol balanceado.

<sup>4</sup>Grow: mediante este método los árboles tienen forma irregular.

<sup>5</sup>Ramped Half-and-Half: mitad y mitad del full y grow, generando mayor diversidad en la población.

- 4 métodos de selección (ruleta, torneo, torneo con parsimonia lexicografica, torneo doble.
- Condiciones de paro de la evolución.

## **LIBSVM**

Es una aplicación para la clasificación de vectores de soporte, regresión y estimación de distribuciones. Soporta la clasificación multiclase (Chang y Lin , 2001). La meta es ayudar a los usuarios especializados en otros campos de investigación a utilizar los SVMs de una manera fácil. LIBSVM provee una interfaz sencilla donde los usuarios pueden ligarla fácilmente a sus propios programas. Las características principales de LIBSVM son:

1. Diferentes formulaciones de SVMs.
2. Clasificación multiclase.
3. Validación cruzada para el método seleccionado.
4. Estimación de probabilidad.
5. Código fuente en varios lenguajes de programación: MATLAB, C++, Java, etc.

# Capítulo VIII

## Conclusiones, aportaciones y trabajo futuro

### Conclusiones

Los resultados experimentales demuestran que el enfoque planteado es capaz de evolucionar la extracción de características a través de la programación genética. En esta investigación se implemento la evolución de la capa S2, donde, se le aplican una serie de operadores matemáticos para resaltar información relevante en la imagen, posteriormente se diseña un mecanismo para obtener los 200 valores máximo en las 6 bandas para la obtención del vector de características.

Por otro lado, con la programación genética nos damos cuenta de que la evolución está íntimamente ligada a la exploración de espacios de búsqueda de soluciones, que no se han contemplado. Por lo tanto el individuo ganador fue capaz de encontrar las operaciones matemáticas para realizar un diccionario de características sin que estuviera de manera explícita en la función de aptitud. También demuestra que la aplicación de esta metodología nos permite potencializar el uso de la extracción de características de una imagen en la clasificación de múltiples clases de objetos.

### Aportaciones

La meta principal de esta tesis era proponer un enfoque basado en el cómputo evolutivo que nos permita estudiar y sintetizar operadores a través de programación genética

(GP), mediante un modelo jerárquico inspirado en las capas V1, V2 y V4 de la corteza visual humana y en el funcionamiento de las células simples y complejas. Esto facilita la extracción de características locales en imágenes digitales pertenecientes a una misma clase de objetos; iniciando en la corteza visual primaria V1 y terminando en la corteza infero temporal donde se lleva a cabo el reconocimiento de los objetos.

Los resultados que se muestran en este trabajo ofrecen tres tipos de contribuciones. Primero, una implementación de un modelo estándar inspirado en la ruta ventral o ruta del ¿qué? Segundo, la aceleración del modelo por medio de la unidad de procesamiento gráfico (GPUs), que aprovecha las capacidades de cálculo en paralelo. Y por último, un enfoque novedoso en la extracción de características locales por medio de programación genética en la ruta ventral a través de un proceso de evolución.

Estos resultado experimentales confirman las conclusiones de nuestro trabajo y demuestra que los resultados son comparables en calidad pero con una eficiencia mayor respecto a aquellos del estado-del-arte.

### **Trabajo futuro**

Como primer trabajo futuro se pretende aumentar la complejidad del número de clases de objetos. Con esta tesis se habren un nuevo panorama en el problema de clasificación de reconocimiento de objetos multiclase y nos permite aterrizar está metodología a diferentes tareas de reconocimiento de objetos, entre ellas:

- Reconocimiento de objetos y localización en una escena en un robot movil autónomo.
- Aplicado a sistemas de recuperación de imágenes basadas en contenido.

- Desarrollar un sistema de reconocimiento de objetos a un nivel cognitivo más alto.
- Combinar el proceso de reconocimiento de objetos con la atención visual ubicado en la ruta dorsal.

Sin embargo, es importante considerar que se puede diseñar un herramienta en CUDA a través de la GPGPU para evolucionar los operadores matemáticos aplicados a las imágenes que nos permitan realizar tareas de reconocimiento de objetos casi en tiempo real. Esto nos permite investigar nuevas técnicas evolutivas para resolver problemas de aprendizaje máquina en menor tiempo de procesamiento.

# Bibliografía

- Agarwal, S. y Roth, D. 2002. "Learning a sparse representation for object detection". Lecture Notes In Computer Science, Proceedings of the 7th European Conference on Computer Vision-Part IV. 2353:113-130 p.
- Akenine-Möller, T. 2002. "Real-time rendering". A.K. Peters Ltd. 3:45 p.
- Allwein, E. L., Schapire, R. E., y Singer, Y. 2001. "Reducing multiclass to binary: a unifying approach for margin classifiers". The Journal of Machine Learning Research. 1:113-142 p.
- Attneave, F. 1954. "Some information aspects of visual perception". Psychol. Rev. 61:183-193 p.
- Banzhaf, W., Nordin, P., Keller, R. E., y Francone, F. D. 1998. "Genetic programming: An introduction: on the automatic evolution of computer programs and its applications". Morgan Kaufmann Publisher. 1:89-95 p.
- Barlow, H. B. y Levick, W. R. 1965. "The mechanism of directionally selective units in rabbit's retina". J. Physiol. 178:477-504 p.
- Brameier, M. 2004. "On linear genetic programming". Universität Dortmund, Germany. 1:125-132 p.
- Branch, J. W. y Olague, G. 2001. "La visión por computadora. una aproximación al estado del arte". International Journal of Computer Vision. 68(PA: 1631):133 p.
- Cagnoni, S., Lutton, E., y Olague, G. 2007. "Genetic and evolutionary computation for image processing and analysis. eurasip book series on signal processing and communications. hindawi publishing corporation". 8:3-8 p.

- Cagnoni, S., Lutton, E., y Olague, G. 2008. "Genetic and evolutionary computation for image processing and analysis", volume 8. EUROSIP Book Series on Signal Processing and Communication. New York, USA. 135-155 p.
- Cajal, S. R. Y. 1899. "Estudios sobre el plan estructural y composición histológica de los centros nerviosos adicionados de consideraciones fisiológicas fundadas en los nuevos descubrimientos. en textura del sistema nervioso del hombre y de los vertebrados", volume 1 Tomo I. Imprenta y librería de Nicolás Moya. Madrid. 288-315 p.
- Caltech101 2001. "Base de datos 101 categories objects". California Institute of Technology.
- Caltech5 2005. "Base de datos seleccionadas de caltech". California Institute of Technology.
- Chang, C.-C. y Lin, C.-J. 2001. "Libsvm: a library for support vector machines".
- CUDA, N. 2007. "Nvidia cuda compute unified device architecture, programming guide".
- Darwin, C. 1993. "The origin of species by means of natural selection: The preservation of favored races in the struggle for life", volume 3. Random House. 85-105 p.
- Dreher, B. 1972. "Hipercomplex cell in the cat's striate cortex". Invest. Ophthalmol. Visc. Sci. 11:335-356 p.
- Duda, H. 2000. "Pattern classification". Wiley 0471056693. 2nd edición. 85-91 p.
- Duda, R. O., Hart, P. E., y Stork, D. G. 2001. "Unsupervised learning and clustering". Pattern classification (2nd edition). 1:571 p.
- Ebner, M. 2009. "A real-time evolutionary object recognition system, proceedings of the 12th european conference on genetic programming (eurogp)". EuroGP. 5481:268-279 p.

- Essen, D. C. V., Newsome, W. T., Maunsell, J. H. R., y Bixby, J. L. 1986. "The projections from striate cortex (v1) to areas v2 and v3 in the macaque monkey: asymmetries, areal boundaries and patchy connections". *J. Comp. Neurol.* 244:451-480 p.
- Fei-Fei, L., Fergus, R., y Perona, P. 2004. "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories". In *Workshop on Generative Model Based Vision.* 1:59-70 p.
- Fergus, R., Perona, P., y Zisserman, A. 2003. "Object class recognition by unsupervised scale-invariant learning". *IEEE CVPR.* 2:246-271 p.
- Fix, E. y Hodges, K. L. 1951. "Discriminatory analysis, nonparametric discrimination: Consistency properties". Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas. 4:261-279 p.
- Fukushima, K. 1980. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biological Cybernetics.* 36:193-202 p.
- Fukushima, K. 1982. "Neocognitron: A new algorithm for pattern recognition tolerant of deformation and shift in position". *Pattern Recognition.* 8(5):1090-1105 p.
- Fung, J., Mann, S., y Aimone, C. 2005. "Openvidia:parallel gpu computer vision". *Proceedings of the 13th annual ACM international conference on Multimedia.* 1:849-852 p.
- Gonzalez, R. C. y Woods, R. E. 2001. "Digital image processing, second edition". Prentice Hall. 2nd edición. 611-619 p.
- Gross, C. G., Rocha-Miranda, C. E., y Bender, D. B. 1972. "Visual properties of neurons in inferotemporal cortex of the macaque". *J. Neurophysiol.* 35:96-111 p.
- Harding, S. y Banzhaf, W. 2008. "Genetic programming on gpu for image processing". *International Journal of High Performance Systems Architecture.* 1:231-240 p.

- Hartley, R. y Zisserman, A. 2003. "Multiple view geometry in computer vision". Cambridge University Press. New York, NY, USA. 672 p.
- Hartline, H. K. 1940. "Receptive field of optic nerve cells". *American Journal of Physiology*. (130):690-699 p.
- Hernández, B., Olague, G., Trujillo, L., y Romero, E. 2007. "Visual learning of texture descriptor for facial expression recognition in thermal imagery". *Computer Vision and Image Understanding*, Elsevier Science. 106:258-269 p.
- Hietanen, J. K. y Perrett, D. I. 1993. "Motion sensitive cells in the macaque superior temporal polysensory area". *Experimental Brain Research*. Springer-Verlag. 93:117-128 p.
- Holland, J. H. 1992. "Adaptation in natural and artificial systems". MIT Press. second edición. 85-92 p.
- Horton, J. y Hedley-Whyte, E. T. 1984. "Mapping of cytochrome oxidase patches and dominance columns in human visual cortex". *Phil. Trans. R. Soc. Lond. B* 304:255-272 p.
- Hubel, D. 1982. "Exploration of the primary visual cortex". *Nature*. 299:515-524 p.
- Hubel, D. y Wiesel, T. 1959. "Receptive fields of single neurones in the cat's striate cortex". *J. Physiol.* 148:574-591 p.
- Hubel, D. y Wiesel, T. 1962. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". *J. Physiol.* 160:106-154 p.
- Hubel, D. y Wiesel, T. 1965. "Receptive fields and functional architecture in two non-striate visual areas (18 and 19) of the cat". *J. Neurophysiol.* 28:229-289 p.
- Hubel, D. y Wiesel, T. 1968. "Receptive fields and functional architecture of monkey striate cortex". *J. Physiol.* 195:215-243 p.

- Hubel, D. y Wiesel, T. 1972. "Laminar and columnar distribution of geniculo-cortical fibers in the macaque monkey". *J. Comp. Neurol.* 146:421-450 p.
- Hubel, D., Wiesel, T., y Levay, S. 1976. "Functional architecture of area 17 in normal and monocularly deprived macaque monkeys". *Cold. Spring. Harbor. Symp. Quant. Biol.* 195:215-243 p.
- Hubel, D. H. 1988. "Eye, brain, and vision". *Scientific American Library.* New York. Cap. 4 p.
- Iwai, E. y Mishkin, M. 1969. "Further evidence on the locus of the visual area in the temporal lobe of the monkey". *Exper. Neurol.* 25:585-594 p.
- Jones, J. P. y Palmer, L. A. 1987. "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex". *J. Neurophys.* 58:1233-1258 p.
- Kandel, E. R., Schwartz, J. H., y Jessell, T. M. 1995. "Essentials of neuronal science and behavior". *Springer Verlag.* Apleton y lange edición. Norwalk. 393 p.
- Koza, J. R. 1989. "Hierchical genetic algorithm operating on populations of computer programs". En *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.* IJCAL-89:768-774 p.
- Koza, J. R. 1992. "Genetic programming: On the programming of computer by means of natural selection". *A Bradford Book.* MIT Press. Cambridge, Massachusetts. London, England. 458-512 p.
- Koza, J. R. 1994. "Scalable learnign in genetic programming using automatic function definition", volume chapter 5 of *A Bradford Book.* MIT Press. Cambridge, Massachusetts. 99-117 p.
- Koza, J. R. 1995. "Survey of genetic algorithms and genetic programming". *Proceedings of 1995. Conference:*589-594 p.

- Koza, J. R., Keane, M., Streeter, M., Mydlowec, W., Yu, J., y Lanza, G. 2003. "Genetic programming iv: Routine human-competitive machine intelligence". Springer. 6(2):221-233 p.
- Kuffler, S. W. 1953. "Discharge pattern and functional organization of mammalian retina". J. Neurophysiol. 16:37-68 p.
- LeCun, Y., Buttou, L., Bengio, Y., y Haffner, P. 1998. "Gradient-based learning applied to document recognition". Proceedings of the IEEE. 86(11):2278-2324 p.
- Li, M. y Vitányi, P. 1997. "An introduction to kolmogorov complexity and its applications". Springer.
- Liu, C. 1995. "A bayesian discriminating features method for face detection". IEEE PAMI. 25(6):725-740 p.
- Livinstone, M. S. y Hubel, D. 1976. "Thalamic inputs to cytochrome oxidase-rich regions in monkey visual cortex". Proc. Natl. Acad. Sci. U.S.A. 79:6098-6101 p.
- Livinstone, M. S. y Hubel, D. 1984. "Specificity of intrinsic connections in primate primary visual cortex". J. Neurosci. 4:2830-2838 p.
- Livinstone, M. S. y Hubel, D. 1987. "Psychophysical evidence for separate channels for the perception of form, color, movement, and depth". J. Neurosci. 7:3416-3468 p.
- Livinstone, M. S. y Hubel, D. 1988. "Segregation of form, color, movement, and depth: anatomy, physiology, and perception". Science. 240:740-749 p.
- Lohmann, V. 1991. "Application of evolution strategy in parallel populations". Proceedings of the 1st Workshop on Parallel Problem Solving from Nature.
- Lund, J. S. 1973. "Organization of neurons in the visual cortex, area 17 of the monkey macaca mulatta". J. Comp. Neurol. 147:455-496 p.
- Marr, D. 1982. "Vision". Freeman. San Francisco.

- McCarthy, J. 1997. "Recursive functions of symbolic expressions and their computation by machine, part i". *Communications of the ACM*. 3 (4):184-195 p.
- Mitchell, M. 1998. "An introduction to genetic algorithm". MIT Press. Cambridge, Massachusetts.
- Mohan, A., Papageorgiou, C., y Poggio, T. 2001. "Example-based object detection in images by components". *IEEE PAMI*. 23(4).
- Movellan, J. R. 2009. "Tutorial on gabor filters".
- Mutch, J. y Lowe, D. 2006. "Multiclass object recognition with sparce, localized feature". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Mutch, J. y Lowe, D. 2008. "Object class recognition and localization using sparse features with limited receptive fields". *International Journal of Computer Vision (IJCV)*.
- Ng, J., Bharath, A., y Zhaoping, L. 2007. "A survey of architecture and function of the primary visual cortex (v1)". *Journal on Applied Signal Processing*.
- Novak, G. 2005. "Automatic programming: Lecture notes, university of texas".
- Olague, G. 2007. "Survey on the state-of-the-art evolutionary computer vision". *Genetic adn Evolutionary Computation Conference*. Londo, United Kingdom; Proceedings of the 2007 GECCO conference companion on Genetic and Evolutionary Computation.
- Perez, C. B. y Olague, G. 2007. "Unsupervised evolutionary segmentation algorithm based on texture analysis". *9th European Worshop on Evolutionary Computation in Image Analysis an Signal Processing (EvoIASP)*. Springer-Verlag.
- Pérez, C. B. y Olague, G. 2008. "Learning invariant region descriptor operators with genetic programming and the f-measure". *International Conference on Pattern Recognition (ICPR)*, Tampa, FL, USA.

- Poli, R., Lagdon, W., y McPhee, N. 2001. "A field guide to genetic programming".
- Ranzato, M. A., Huang, F., Boreau, Y., y LeCun, Y. 2007. "Unsupervised learning of invariant feature hierarchies with application to object recognition". In Proc. Computer Vision and Pattern Recognition Conference (CVPR).
- Riesenhuber, M. y Poggio, T. 1999. "Hierarchical models of object recognition in cortex". *Nature neuroscience*. 2(11):1019-1025 p.
- Rodamn, H. R., Gross, C. G., y Albright, T. D. 1989. "Afferent basics of visual response properties in area mt of the macaque: I. effects of striate cortex removal". *J. Neurosci.* 9:2033-2050 p.
- Roy, S. D., Chaudhury, S., y Banerjee, S. 2004. "Active recognition through next view planning: a survey". *Pattern Recognition*.
- Schneiderman, H. y Kanade, T. 2000. "A statistical approach to 3d object detection applied to faces and cars". *Proc. CVPR*.
- Schölkopf, B. y Smola, A. J. 2001. "Learning with kernels: Support vector machines, regularization, optimization, and beyond". MIT Press, Cambridge, MA, USA.
- Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., y Poggio, T. 2005a. "A theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex". Massachusetts institute of technology-computer science and artificial intelligence laboratory.
- Serre, T., L.Wolf, Bilechi, S., Riesenhuber, M., y Poggio, T. 2007a. "Robust object recognition with cortex-like mechanisms". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 29(3):411-426 p.
- Serre, T., Oliva, A., y Poggio, T. 2007b. "A feedforward architecture accounts for rapid categorization". Center for Biological and Computational Learning, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. 15:6424-6429 p.

- Serre, T., Wolf, L., y Poggio, T. 2004. "A new biologically motivated framework for robust object recognition". CBCL Memo 243.
- Serre, T., Wolf, L., y Poggio, T. 2005b. "Object recognition with features inspired by visual cortex". *Journal on Applied Signal Processing*.
- Shawe-Taylor, J. y Cristianini, N. 2004. "Kernel methods for pattern analysis". Cambridge University Press, New York, USA.
- Sinha, S., Franhm, J., y Pollefeys, M. 2006. "Gpu-based video feature tracking and matching". In *Workshop on Edge Computing Using New Commodity Architectures*.
- Smith, S. 1980. "A learning system based on genetic adaptive algorithms". Tesis de Doctorado, University of Pittsburgh.
- Spall, J. 2003. "Introduction to stochastic search and optimization". John Wiley and Sons. Cambridge, Massachusetts.
- Tootell, R. R. H., Silverman, M. S., Valois, R. L. D., y Jacobs, G. 1983. "Functional organization of the second cortical visual area in primates". *Science*. 220:737-739 p.
- Trujillo, L. y Olague, G. 2006. "Synthesis of interest point detectors though genetic programmin". In *GECCO '06: Proceedings of the 8th annual conference on Genetic and Evolutionary Computation*, Seattle, Washintonn, USA, ACM Press.
- Trujillo, L. y Olague, G. 2008. "Automated design of image operators that detect interest points". MIT Press. 16(4):483-507 p.
- Ungerleider, L. G. y Mishkin, M. 1982. "two cortical visual systems". in *analysis of visual behaviour*. Cambridge, Mass.: MIT Press.
- Vapnik, V. N. 1995. "The nature of statistical learning theory". Springer.
- Viola, P. y Jones, M. 2001. "Rapid object detection using a boosted cascade of simple features". *Proc. CVPR*. 1:511-518 p.

- Wloka, M. y Huddy, R. 2003. "Directx9 performance.where does it como from, and where does it all go?". Game Developers Conference.
- Wong-Riley, M. 1989. "Reciprocal connections betwen striate and prestriate cortex in squirrel monkeys as demonstrated by combined peroxidase histochemistry and autoradiography". Brain. 147:159-164 p.
- Zeki, S. 1980. "The representation of colours in the cerebral cortex". Nature. 284:412-418 p.
- Zeki, S. 1989. "Functional specialization in the visual cortex of the rhesus monkey". Nature. 147:423-428 p.
- Zeki, S. y Shipp, S. 1988. "The functinoal logic of cortical connections". Nature. 335:311-317 p.