



# INSTITUTO POLITÉCNICO NACIONAL

---

---

## CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Metodologías para la reducción del número de imágenes  
requeridas para el reconocimiento de objetos articulados

### T E S I S

QUE PARA OBTENER EL GRADO DE:

DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

M.C. LUIS CARLOS ALTAMIRANO ROBLES

DIRECTORES DE TESIS:

DR. LEOPOLDO ALTAMIRANO ROBLES <sup>+</sup>

DR. JOSÉ MATÍAS ALVARADO MENTADO <sup>++</sup>

<sup>+</sup> Investigador del INAOE - Puebla.

<sup>++</sup> Investigador del CIC - IPN.



MÉXICO, D.F., ENERO DE 2002.

**A MIS PADRES:**

**GRACIELA ROBLES PÉREZ**

**LEOPOLDO ALTAMIRANO PÉREZ**

# Agradecimientos

Primeramente quiero agradecer a mis directores de tesis, Dr. José Matías Alvarado Mentado y Dr. Leopoldo Altamirano Robles, por la oportunidad que me brindaron de trabajar bajo su dirección, así como por su acertada asesoría, comprensión y paciencia durante todo el desarrollo de este trabajo doctoral.

Posteriormente quiero agradecer a mis sinodales, Dr. Ángel Fernando Kuri Morales, Dr. Juan Humberto Sossa Azuela, Dr. Serguei Levachkine y Dr. Juan Luis Díaz de León Santiago, por sus acertados comentarios y sugerencias que sirvieron para guiar y mejorar desde sus inicios hasta la culminación a este trabajo doctoral.

Al Centro de Investigación en Computación y al Instituto Politécnico Nacional, por darme la oportunidad de realizar mis estudios doctorales dentro de sus instalaciones.

Al Instituto Nacional de Astrofísica, Óptica y Electrónica por permitirme hacer uso de sus instalaciones para la realización de gran parte de mi trabajo doctoral.

A la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla, por la preparación que me brindó a lo largo de mis estudios de licenciatura y maestría. En particular quiero agradecer el apoyo moral de su actual director, Dr. Guillermo de Ita Luna.

A Margarita Carrasco por su gran apoyo, comprensión y colaboración a lo largo de todo el desarrollo de mis estudios de posgrado.

A mi familia entera por su apoyo moral para el desarrollo y culminación de mis estudios doctorales. En particular, a mis hermanas Rosa Elena, Elvia y Ana Luisa, a mi hermano Leopoldo, y de manera especial a mis sobrinos: Jorge Eduardo, Armando, Alfredo, Aída Graciela, Carlitos, Angélica, Verónica y Juan Manuel, que con tan sólo su compañía hicieron menos difícil el desarrollo de mis estudios de posgrado.

A mis amigos: Luis Flores, Pedro Bello y Luis Rojas, porque aunque la vida nos ha llevado por caminos distintos, la amistad aún continúa.

Finalmente, a la Dra. Mariofanna Milanova por el gran apoyo y confianza que me brindó al inicio de mis estudios doctorales, y durante todo el tiempo que permaneció en el CIC. Al Dr. Leonid Sheremetov y a Soila Vargas por sus acertadas revisiones de manuscritos de este trabajo doctoral. Al Dr. Álvaro de Albornoz Bueno por sus acertados consejos, sugerencias y apoyo durante toda mi estancia en el CIC.

# Índice

Índice .....	I
Glosario de términos .....	IV
Índice de Figuras.....	V
Índice de Tablas .....	VIII
INTRODUCCIÓN .....	1
<b>1 TRABAJOS PREVIOS SOBRE RECONOCIMIENTO DE OBJETOS USANDO MODELOS BASADOS EN APARIENCIA.....</b>	<b>4</b>
<b>PARTE I:</b> Modelos basados en apariencia .....	4
I.1 Definición y proceso de reconocimiento .....	8
I.2 Creación de modelos.....	11
I.2.1 Compresión.....	14
I.2.2 Interpolación .....	15
I.2.3 Modelos Híbridos.....	16
I.3 Procesamiento de la escena.....	16
I.4 Comparación ( <i>matching</i> ) .....	17
I.5 Verificación de hipótesis .....	19
<b>PARTE II:</b> Enfoques geométricos vs. basados en apariencia.....	20
II.1 Enfoques geométricos.....	20
II.2 Comparación entre enfoques geométricos y basados en apariencia ..	21
<b>PARTE III:</b> Aplicaciones de los modelos basados en apariencia .....	23
III.1 Posicionamiento de robots.....	23
III.2 Reconocimiento de ademanes .....	24
III.3 Reconocimiento de rostros .....	25
<b>PARTE IV:</b> Problemática a resolver .....	27
<b>2 PRELIMINARES MATEMÁTICOS.....</b>	<b>31</b>
2.1 Definiciones .....	31
2.2 Adquisición uniforme de imágenes.....	33
2.3 Reconocimiento de objetos usando espacios propios ( <i>eigenspaces</i> ) ...	36
<b>3 MODELADO Y RECONOCIMIENTO MEJORADO DE OBJETOS.....</b>	<b>38</b>
3.1 Requerimientos de almacenamiento .....	38
3.2 Requerimientos de tiempo de procesamiento .....	39
3.3 Reducción de la cantidad de imágenes .....	40

<b>4 ADQUISICIÓN NO-UNIFORME DE IMÁGENES .....</b>	<b>42</b>
4.1 Adquisición adaptativa (guiada por interpolación lineal y SSD).....	42
4.2 Algoritmo adaptativo .....	45
4.3 Muestreo no-uniforme y espacios propios.....	48
<b>5 GENERALIZACIÓN PARA N-PARÁMETROS.....</b>	<b>50</b>
5.1 Adquisición de imágenes .....	51
5.2 Interpolación N-lineal.....	51
5.3 Adquisición adaptativa de imágenes generalizada.....	53
5.4 Algoritmo generalizado para N-parámetros .....	55
5.5 Consideraciones sobre el algoritmo generalizado propuesto.....	55
<b>6 RESULTADOS EXPERIMENTALES .....</b>	<b>56</b>
6.1 Un parámetro.....	56
6.1.1 Objetos de estudio.....	56
6.1.2 Determinación del número de imágenes requeridas .....	58
6.1.3 Determinación de las imágenes estrictamente necesarias ...	59
6.1.4 Distribución de las imágenes estrictamente necesarias.....	62
6.1.5 Precisión de los modelos .....	66
6.1.6 Reducción de la cantidad de imágenes .....	76
6.2 Múltiples parámetros.....	78
6.2.1 Objetos de estudio .....	79
6.2.2 Determinación del número de imágenes requeridas.....	79
6.2.3 Distribución de las imágenes estrictamente necesarias.....	80
6.2.4 Precisión de los modelos .....	82
6.2.5 Reducción de la cantidad de imágenes.....	82
<b>7 APLICACIONES .....</b>	<b>85</b>
7.1 Descripción.....	85
7.2 Trabajo previo.....	86
7.3 Justificación del empleo de la propuesta .....	86
7.4 Resultados experimentales.....	86
7.4.1 Objetos de estudio .....	87
7.4.2 Parámetros empleados .....	87
7.4.3 Adquisición adaptativa de imágenes .....	87
7.4.4 Precisión de los modelos.....	89
7.4.5 Reducción de la cantidad de imágenes.....	90
7.4.6 Reconocimiento .....	91
<b>8 CONSIDERACIONES FINALES .....</b>	<b>92</b>
8.1 Detección y corrección de errores en el modelo.....	92
8.1.1 Correctez y completitud del proceso propuesto.....	96

8.1.2 Resultados experimentales .....	98
8.2 Determinación automática de precisión.....	105
8.2.1 Resultados experimentales .....	107
8.3 Empleo de interpolación no-lineal en la propuesta .....	109
8.4 Selección de un enfoque (¿geométrico ó basado en apariencia?) .....	117
<b>9 CONCLUSIONES Y APORTACIONES .....</b>	<b>119</b>
<b>APÉNDICE A: Publicaciones .....</b>	<b>123</b>
<b>APÉNDICE B: Proyectos de investigación .....</b>	<b>125</b>
<b>APÉNDICE C: Algoritmo generalizado para N parámetros.....</b>	<b>126</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>128</b>

# Glosario de términos

<b>Imagen</b>	Representación funcional obtenida por un sensor, de los patrones de intensidad de alguna forma de energía en una porción del espacio.
<b>Imagen digital</b>	Imagen que ha sido discretizada espacialmente y cuantificada en niveles de gris ó color.
<b>Interpolación</b>	Técnica empleada para estimar valores desconocidos de una función, a partir de un conjunto de valores conocidos dentro de un intervalo de valores de interés.
<b>Modelo basado en apariencia</b>	Representación de un objeto físico, obtenida al adquirir un conjunto de imágenes alrededor del objeto.
<b>Sensor</b>	Dispositivo capaz de registrar patrones de intensidad de alguna forma de energía en una porción del espacio.

# Índice de Figuras

Figura 1	(a) un objeto. (b) posible forma de modelar el objeto en (a) a través de su contorno. Nótese que la información en forma de texto e imágenes grabadas sobre el objeto ya se ha perdido.	5
Figura 2	Dos objetos (izquierda) y sus posibles modelos geométricos correspondientes (derecha), modelados a través de sus contornos.	5
Figura 3	Sistema para la adquisición de imágenes. La fuente de iluminación y la cámara pueden tomar cualquier posición con respecto al objeto.	6
Figura 4	Componentes típicos de un sistema de reconocimiento que emplea modelos basados en apariencia.	10
Figura 5	Conjunto de objetos sobre los que se ha experimentado SLAM: Software Library for Appearance Matching.	22
Figura 6	Objeto empleado para adquirir la secuencia de imágenes mostrada en la Figura 7.	34
Figura 7	Adquisición de imágenes para el objeto mostrado en la Figura 6. El ángulo de giro de la tornamesa entre imágenes consecutivas es de $10^\circ$ .	35
Figura 8	Un objeto para el cual la reducción en la cantidad de imágenes es extrema: una pelota.	42
Figura 9	Objetos que presentan simetría central respecto al centro de la tornamesa.	43
Figura 10	La Figura (a) muestra el comportamiento de la trayectoria determinada por la apariencia del objeto. Aunque la trayectoria realmente reside en un espacio multidimensional, aquí se muestra en un plano para propósitos ilustrativos. En (b) se muestra la interpolación lineal segmentaria para la trayectoria de (a). La diferencia entre la apariencia del objeto (curva oscura) y la estimación por interpolación lineal (curva en líneas claras) es menor que $\epsilon$ en los puntos medios de cada línea recta. En (c) se muestra el comportamiento estimado por medio de la técnica propuesta. Los puntos de las esquinas señalan a las imágenes estrictamente necesarias para aproximar la apariencia del objeto, dentro de un error $\epsilon$ dado.	44
Figura 11	Ejemplo de interpolación de imágenes, mediante el empleo de la expresión (1) aplicada a las imágenes 1 y 2 ( $g(0.5)$ ). La imagen real para el punto correspondiente es también mostrada.	45
Figura 12	Una vista esquemática de cómo el algoritmo propuesto logra aproximar la apariencia del objeto, mediante interpolación lineal segmentaria (ver texto para explicación).	47
Figura 13	La interpolación bilineal interpola entre 4 puntos, por medio de la superficie más simple entre ellos: un paraboloides hiperbólico (adaptado de [Fa88]).	52
Figura 14	Algunos objetos propios empleados para experimentar el algoritmo 1.	57



Figura 15	Objetos que forman a COIL-20 (Columbia Object Image Library).	57
Figura 16	Distribución sobre el plano de la tornamesa de las imágenes estrictamente necesarias para modelar la apariencia de los objetos de la Figura 14.	64
Figura 17	Graficación de la distribución de las imágenes estrictamente necesarias, para capturar la apariencia de los objetos que forman a COIL-20 (Figura 15).	65
Figura 18	Análisis de la precisión de distintos modelos, generados mediante la técnica propuesta, para el objeto 6 de la Figura 14, en función de distintos valores para la precisión $\epsilon$ . Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes.	67
Figura 19	Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de la Figura 14, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión $\epsilon$ señalada. Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes.	70
Figura 20	Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de COIL-20 (Figura 15), empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión $\epsilon$ señalada. Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes.	72
Figura 21	Un ejemplo de la forma de adquirir imágenes para un objeto, para construir su modelo bajo 2 parámetros, empleando un muestreo uniforme con $45^\circ$ de incremento en el ángulo de giro de la tornamesa y $5^\circ$ de incremento en el ángulo de elevación de la cámara, respecto al plano de la tornamesa, entre imágenes consecutivas.	78
Figura 22	Distribución de las imágenes necesarias para modelar la apariencia de los objetos de la Figura 14, bajo 2 parámetros: rotación de la tornamesa (eje X) y elevación de la cámara (eje Y).	81
Figura 23	Pernos industriales estudiados.	87
Figura 24	Algunas posiciones en que puede aparecer un perno soldado a una carrocería.	88
Figura 25	Distribuciones de las imágenes requeridas para modelar los pernos estudiados.	88
Figura 26	Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los pernos de la Figura 23, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión $\epsilon$ señalada. Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes.	89

Figura 27	Graficado de los resultados obtenidos al estimar la posición de los objetos mostrados en la Figura 23 en 72 imágenes por objeto. La estimación de posición será precisa si la posición real es igual a la estimada por el sistema, por tanto, una línea recta en una gráfica denota una estimación precisa para todas las imágenes. Eje X: posición real del objeto. Eje Y: posición estimada por el sistema.	90
Figura 28	Los 3 casos en que se pueden presentar las zonas imprecisas.	94
Figura 29	Diagrama de bloques que describe el proceso predictor-corrector propuesto, para la creación del modelo de un objeto, que cumple con la restricción de precisión $\epsilon$ , en todos los puntos contenidos dentro de un intervalo inicial dado, y que por tanto, no contiene ninguna zona imprecisa.	95
Figura 30	Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de la Figura 14, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión $\epsilon$ señalada. Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes. Derecha: Modelo generado mediante el algoritmo 1 (Capítulo 4). Izquierda: Modelo corregido mediante el proceso predictor-corrector (Figura 29). Obsérvese la precisión en todas las zonas del modelo corregido. Solamente los objetos que requirieron corrección son mostrados.	100
Figura 31	Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de la Figura 15, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión $\epsilon$ señalada. Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes. Derecha: Modelo generado mediante el algoritmo 1 (Capítulo 4). Izquierda: Modelo corregido mediante el proceso predictor-corrector (Figura 29). Obsérvese la precisión en todas las zonas del modelo corregido. Solamente los objetos que requirieron corrección son mostrados.	103
Figura 32	Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto empleando no-linealidad y la apariencia real de los objetos de la Figura 14, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión $\epsilon$ señalada. Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes.	112
Figura 33	Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto empleando no-linealidad y la apariencia real de los objetos de la Figura 15, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión $\epsilon$ señalada. Las partes del gráfico bajo la precisión $\epsilon$ denotan buenas aproximaciones, y las partes sobre la precisión $\epsilon$ indican que se requiere mejorar el modelo del objeto en esas partes.	114

# Índice de Tablas

Tabla 1	Tabulación del número de imágenes requeridas para capturar la apariencia de un objeto, en función del número de parámetros.	39
Tabla 2	Número de imágenes requeridas para capturar la apariencia de los objetos mostrados en la Figura 14, dependiendo de distintos valores de $\epsilon$ . Las celdas marcadas con + indican que para alcanzar la precisión $\epsilon$ señalada, el ángulo de rotación de la tornamesa entre imágenes sucesivas debe ser inferior a $5^\circ$ .	58
Tabla 3	Número de imágenes requeridas para capturar la apariencia de los objetos en COIL-20 (Figura 15), dependiendo de distintos valores de $\epsilon$ . Las celdas marcadas con + indican que para alcanzar la precisión $\epsilon$ señalada, el ángulo de rotación de la tornamesa entre imágenes sucesivas debe ser inferior a $5^\circ$ .	59
Tabla 4	Análisis de la precisión de los objetos mostrados en la Figura 14.	60
Tabla 5	Análisis de la precisión de COIL-20 (Figura 15).	61
Tabla 6	Tabla comparativa entre un muestreo uniforme y el muestreo no-uniforme generado por la propuesta, respecto al número de imágenes requeridas para los objetos de la Figura 14.	76
Tabla 7	Tabla comparativa entre un muestreo uniforme y el muestreo no-uniforme generado por la propuesta, respecto al número de imágenes requeridas para los objetos de la Figura 15 (COIL-20).	77
Tabla 8	Número de imágenes requeridas para capturar la apariencia de los objetos en la Figura 14, dependiendo de distintos valores de $\epsilon$ , y empleando 2 parámetros (rotación de la tornamesa y elevación de la cámara). Las celdas marcadas con + indican que para alcanzar la precisión fijada por $\epsilon$ , el ángulo de rotación de la tornamesa ó el ángulo de elevación de la cámara entre imágenes sucesivas, debe ser inferior a $5^\circ$ .	79
Tabla 9	Tabla comparativa entre un muestreo uniforme y el muestreo no-uniforme generado por la propuesta, respecto al número de imágenes requeridas para los objetos de la Figura 14, empleando 2 parámetros.	83
Tabla 10	Resultados obtenidos al aplicar el proceso predictor-corrector a los objetos de la Figura 14.	99
Tabla 11	Resultados obtenidos al aplicar el proceso predictor-corrector a los objetos de la Figura 15.	99
Tabla 12	Resultados obtenidos al aplicar las 2 estrategias propuestas, para calcular automáticamente un valor para la precisión $\epsilon$ con el cual, el sistema construirá el modelo de cada uno de los objetos de la Figura 14.	107

Tabla 13	Resultados obtenidos al aplicar las 2 estrategias propuestas, para calcular automáticamente un valor para la precisión $\varepsilon$ con el cual, el sistema construirá el modelo de cada uno de los objetos de la Figura 15.	108
Tabla 14	Comparativo entre interpolación lineal y no-lineal respecto a la máxima precisión obtenida y el número de imágenes requeridas para modelar a los objetos de la Figura 14.	110
Tabla 15	Comparativo entre interpolación lineal y no-lineal respecto a la máxima precisión obtenida y el número de imágenes requeridas para modelar a los objetos de la Figura 15.	110

# Resumen

Los enfoques basados en apariencia surgieron como una alternativa a los enfoques geométricos tradicionales para el reconocimiento de objetos. A diferencia de los enfoques geométricos, los enfoques basados en apariencia no se restringen al empleo únicamente de la geometría de los objetos; en consecuencia, con ellos es posible lograr elevados índices de reconocimiento en aplicaciones específicas, aún entre objetos muy similares entre sí.

Los enfoques basados en apariencia modelan al objeto mediante un conjunto de imágenes de entrenamiento, que en conjunto describen la apariencia del objeto, y que son empleadas posteriormente para reconocerlo. Estas imágenes son adquiridas rotando al objeto frente al sensor, comenzando desde una posición inicial arbitraria, y usualmente, rotando al objeto en un incremento constante en el ángulo de rotación, para adquirir imágenes sucesivas. Esta forma de adquirir las imágenes de entrenamiento genera un muestreo uniforme de la apariencia del objeto, y es independiente del tipo de objeto que se esté modelando. Aunque simple, el empleo de un muestreo uniforme supone una gran capacidad de espacio de almacenamiento y tiempo de cómputo, en los sistemas de reconocimiento que emplean modelos basados en apariencia. Esto se debe a que usualmente se requiere que el incremento del ángulo de rotación entre imágenes sucesivas sea pequeño, para representar apropiadamente la apariencia de cualquier objeto en general.

En esta tesis doctoral se explora el empleo del muestreo no-uniforme al adquirir imágenes de entrenamiento, para generar automáticamente modelos basados en apariencia. Con el muestreo no-uniforme se ha logrado obtener una reducción significativa, de la cantidad de imágenes requeridas para construir modelos basados en apariencia. Adicionalmente, ha sido posible obtener mejores modelos que los obtenidos mediante un muestreo uniforme. Esto ha sido posible gracias a que al realizar un análisis del comportamiento de la apariencia de cada objeto particular, se adquiere una cantidad mayor de imágenes en zonas donde la apariencia varía intensamente, y pocas imágenes donde ésta varía poco. Por ello, la cantidad de imágenes depende tanto de las características específicas de cada objeto particular que se esté modelando (geometría, textura, etc.), como también de un parámetro  $\epsilon$ , que establece la precisión deseada para el modelo.

El análisis es realizado mediante una técnica simple propuesta, que empleando un esquema de interpolación lineal, analiza la apariencia del objeto y a la par, construye su modelo. Este tipo de muestreo ha sido aplicado a una base extensa de imágenes de objetos, mostrando los beneficios de su empleo (disminución del espacio de almacenamiento, del tiempo de cómputo y mejor precisión en los modelos).

Finalmente, en esta tesis doctoral se plantea la forma en que el muestreo no-uniforme propuesto, ha sido incorporado a la técnica de espacios propios para soportar el reconocimiento de objetos, haciendo a ésta más eficiente y preservando la precisión del modelo previamente construido, después de su aplicación.

# Abstract

The appearance-based approaches emerged as an alternative to the traditional geometrical ones for object recognition. Different to those traditional, the appearance-based approaches are not restricted to the only use of the object's geometry; that's why, using them, it is possible to obtain high recognition rates in some specific applications, even among very similar objects.

The appearance-based approaches model the object through a set of training images, which together describe the object's appearance, and afterwards they are used to recognize the object. These images are acquired by rotating the object in front of the visual sensor. Beginning from a fixed initial position, the following new images are captured by rotating the object always in the same angle with respect to the current position. This way of acquiring training images generates a uniform sampling of the object's appearance, and the sampling is independent to the type of object that is being modeled. Although uniform sampling is simple to do, the recognition systems that use appearance-based models need a great storage capacity and computing time, because the increase of the rotation angle among the consecutive images is required to be small to appropriately represent the appearance of any object.

In this PhD. thesis, it is introduced the usage of non-uniform sampling for acquiring an image set, in order to generate automatically appearance-base models. With this non-uniform sampling usage, it has been achieved a significant reduction in the quantity of the necessary images to build appearance-based models, as well as to obtain improved models than those obtained with a uniform sampling. It has been possible, due to the appearance behavior analysis for each particular object, in order to acquire a larger quantity of images in those zones where the object's appearance changes quickly, and few images where it changes slowly. Therefore, the quantity of images depends on specific characteristics of each particular object that is being modeled (geometry, texture, etc.), as well as a desired precision  $\epsilon$  for the model.

This analysis is done through a simple proposed technique, which analyses the object's appearance, and alongside, builds its model by the use of a liner interpolation scheme. This non-uniform sampling has been applied to a large images base, showing the benefits of its usage (diminish storage usage, diminish computing time and better approximation of the object's appearance).

Finally, in this PhD. thesis, it is exposed how non-uniform sampling has been used in conjunction with the eigenspaces technique, making this technique more efficient for object recognition, as well as preserving the precision of the previous built model after its application.

# INTRODUCCIÓN

El empleo de modelos basados en apariencia, es un enfoque propuesto a principios de la década de los 90's para realizar el reconocimiento de objetos [Sh94]. El modelado y reconocimiento basado en la apariencia, emplea información proveniente directamente de imágenes tomadas al objeto, y que en conjunto describen la apariencia de éste (geometría, textura, color, etc.). Este enfoque difiere de los enfoques geométricos [Mu96], [Po94], en que no se restringe al empleo de la geometría del objeto para modelarlo y reconocerlo. La información de apariencia puede ser restringida al empleo de algunas características encontradas en las imágenes (llegando incluso al empleo de solamente el contorno, o de propiedades provenientes de la geometría del objeto, lo que sugiere que la apariencia es una generalización de algunos enfoques geométricos. Ver, por ejemplo los trabajos de Nelson [Ne97], [Ne98a], [Ne98b], [Ne98c]), o bien, puede ser tan general como el empleo de las imágenes completas, para realizar el modelado y el reconocimiento del objeto [Sh94], [Mu92].

Diversas técnicas han sido propuestas para emplear la apariencia en el modelado y reconocimiento de objetos [Sh94], [Tu91], [Hu98], [Mu95], [Po96], [Ne98a]. Una de las primeras técnicas propuestas dentro del enfoque basado en apariencia, fue la hecha por Murase y Nayar [Mu92], [Mu95], basada en el concepto de espacios propios. Esta técnica hace uso de toda la información que contienen las imágenes que modelan la apariencia del objeto, y mediante la obtención de un conjunto ortonormal de vectores propios (*eigenvectors*) y del empleo de éstos para la proyección de las imágenes del objeto a un espacio multidimensional (*eigenspace*), se consigue una reducción significativa del número de imágenes que se emplean para describir la apariencia del objeto. Esta técnica ha mostrado el potencial que el reconocimiento basado en apariencia posee [Mu96], permitiendo modelar y reconocer objetos arbitrariamente complejos en su geometría, y que por tanto, son difíciles de modelar mediante los enfoques geométricos.

Sin embargo, a pesar de que en teoría es posible mediante el empleo de modelos basados en apariencia, realizar el modelado y reconocimiento de objetos cualesquiera que sean éstos, en la práctica modelar y reconocer objetos bajo condiciones variables de iluminación y posición presenta algunos inconvenientes. En este caso se incrementa grandemente la cantidad de imágenes requeridas para describir la apariencia de los objetos, aumentando, en consecuencia, la cantidad de espacio de almacenamiento, así como la cantidad de tiempo de cómputo necesario para construir el modelo. Tal incremento en la cantidad de imágenes, se debe, por una parte, al incremento de los parámetros requeridos, para describir la apariencia del objeto. Por

otra parte, la cantidad de imágenes es considerable debido a que el proceso de adquisición de imágenes no ha sido estudiado adecuadamente, y *erróneamente* se ha supuesto que una cantidad *fija* de imágenes es suficiente para capturar de una manera precisa la apariencia de cualquier objeto. Sin embargo, esto es incorrecto como se señala en [Al00a], [Al00b], [Al02a], [Al02b], puesto que para objetos con simetría central (por ejemplo, una esfera pintada homogéneamente, recipientes cilíndricos, etc.), se requiere generalmente una cantidad menor de imágenes para su reconocimiento, y para objetos con una geometría y/o textura complejas, esta cantidad puede incrementarse, según los requerimientos de precisión que requiera el investigador.

Más aún, el suponer que una cantidad fija de imágenes captura de manera precisa la apariencia de cualquier objeto, repercute en un conocimiento incierto acerca de la precisión de los modelos así obtenidos, y como consecuencia, en un bajo rendimiento de los sistemas de reconocimiento. Se sabe que, en general, aumentar la cantidad de imágenes repercutirá en una mejor calidad de los modelos, pero hacer esto supone una gran capacidad de almacenamiento y cómputo de los sistemas de reconocimiento, en especial en técnicas que realizan un cálculo intenso, como es el caso de la técnica de espacios propios (*eigenspaces*) [Mu92], [Mu95], [Na96a]. En particular, tal técnica se hace prohibitiva cuando se incrementan los parámetros para la tarea de reconocimiento a realizar, y sólo puede ser reducido si se reduce la cantidad de imágenes para un objeto dado.

Para resolver los problemas anteriores, este trabajo doctoral se plantea alcanzar los siguientes

Objetivos principales del presente trabajo doctoral:

- ❖ Mejorar la precisión de los modelos de los objetos.
- ❖ Reducir, *significativamente*, la cantidad de imágenes requeridas para describir la apariencia del objeto. En consecuencia:
  - Reducir el espacio de almacenamiento, y
  - Reducir el tiempo de cómputo requerido para construir el modelo del objeto.

Se propone cumplir los dos principales objetivos anteriores, mediante un proceso de adquisición de imágenes guiado por un método de interpolación lineal, que reducirá la cantidad de imágenes en las zonas del modelo que pueden ser aproximadas por interpolación lineal, y mejorará la precisión en las zonas donde la aproximación es no lineal. Esta forma de adquirir las imágenes para generar el modelo de un objeto, genera un muestreo no-uniforme del espacio visual de trabajo, y es dependiente de cada objeto en particular que se esté modelando, en contraste con las diversas técnicas propuestas en la literatura, y que emplean un muestreo uniforme. Este conjunto reducido de imágenes será proporcionado a la técnica de espacios propios, - la cual



será empleada como soporte para el reconocimiento de los objetos -, para obtener una reducción aún mayor en la cantidad de imágenes.

El empleo del muestreo no-uniforme propuesto ha mostrado en trabajos experimentales [AI00a], [AI00b], [AI02a], [AI02b], que la reducción de la cantidad de imágenes es significativa y la precisión de los modelos generados de esta manera se mejora considerablemente. En consecuencia, se ha logrado reducir tanto el tiempo de procesamiento necesario para construir los modelos, como el espacio de almacenamiento requerido, además de lograr un reconocimiento más preciso, en especial, entre objetos muy similares.

Estos beneficios son obtenidos debido a que utilizando la técnica propuesta se adquieren, solamente, las imágenes estrictamente necesarias de los objetos a ser modelados, bajo una precisión  $\epsilon$  preestablecida. En consecuencia, cuando el objeto es poco complejo en su apariencia (por ejemplo, recipientes cilíndricos), la técnica propuesta es capaz de registrar este hecho y, en consecuencia, adquirir pocas imágenes. En contraste, cuando el objeto es complejo en su apariencia, la técnica registra este hecho, y es capaz de generar un modelo que cumple con la restricción de precisión  $\epsilon$  preestablecida. En este caso, la reducción del número de imágenes puede no ser tan significativa, pero la precisión del modelo obtenido con la técnica propuesta, es superior a la obtenida con un muestreo uniforme [AI02a], [AI02b].

Este documento se encuentra estructurado de la siguiente manera: En el Capítulo 1 se expone una reseña de trabajos previos sobre reconocimiento de objetos, empleando modelos basados en apariencia. En el Capítulo 2 se exponen conceptos preliminares. En el Capítulo 3 se expone detalladamente el problema que se plantea resolver. En el Capítulo 4 se expone la estrategia de solución que se propone. En el Capítulo 5 se plantea la generalización para  $N$  parámetros, de la estrategia de solución planteada en el Capítulo 4. En el Capítulo 6 se exponen los resultados experimentales obtenidos y una discusión sobre las ventajas, limitaciones y repercusiones en otros trabajos, derivadas de la presente propuesta. En el Capítulo 7 se expone la aplicación de la propuesta a un problema de tipo industrial. En el Capítulo 8 se exponen algunas consideraciones finales sobre la propuesta realizada. Finalmente, en el Capítulo 9 se exponen las conclusiones sobre la propuesta realizada, además de las posibles aplicaciones y extensiones que podrían realizársele, junto con las principales aportaciones hechas.

# CAPÍTULO 1

## ***TRABAJOS PREVIOS SOBRE RECONOCIMIENTO DE OBJETOS, USANDO MODELOS BASADOS EN APARIENCIA***

**Resumen:** En este Capítulo se expone una reseña del estado del arte sobre reconocimiento de objetos, empleando modelos basados en apariencia. Sin pretender ser exhaustiva, esta reseña trata de incluir los trabajos más relevantes en el área de interés.

---

### **PARTE I: Modelos basados en apariencia**

El término “visión computacional” (*computer vision*) denota la capacidad que el hombre trata de conferirle a la computadora, para hacer posible que ésta pueda reconocer objetos en cualquier imagen que le sea presentada, tanto o más eficientemente que la forma en que el humano lo hace. El lograr el reconocimiento de objetos a través de imágenes, tiene repercusiones inmediatas en diversas ramas del conocimiento, como en Inteligencia Artificial, en particular robótica, y en tecnologías como el control de calidad y la manufactura de productos, así como en el armado y conducción automática de automóviles, por citar sólo algunas.

Sin embargo, debido a que las leyes que rigen al mundo visual no son fáciles de describir, la tarea de lograr que la computadora realice el proceso de reconocimiento se vuelve extremadamente difícil. Por supuesto, las imágenes pueden ser pensadas como mapas definidos sobre dominios bidimensionales, y se comprende el proceso de formación de la imagen de la interacción física entre superficies, volúmenes y luz, pero el “vocabulario” usado por el cerebro para analizar esta entrada visual, esto es, las representaciones visuales, permanecen aún extremadamente inaccesibles. Esto hace la tarea de construcción de sistemas de visión artificial diferente de otras tareas de Inteligencia Artificial, tales como el procesamiento de lenguaje natural. Esto es también un lado fascinante de la visión computacional, debido a que somos libres para proponer la representación visual más apropiada para una tarea dada.

Uno de los problemas centrales dentro de la visión computarizada, es la construcción de modelos de objetos que puedan ser usados por la computadora, para

poder llevar a cabo la comparación de éstos con patrones que son extraídos de una imagen, y así poder realizar la tarea de reconocimiento. Tradicionalmente, estos modelos se han construido a través de las propiedades geométricas de los objetos, dando lugar a los llamados modelos geométricos; así mismo, se ha hecho uso de descriptores invariantes para representar a los objetos. Este tipo de representaciones han sido ampliamente investigadas, sin embargo, el reconocimiento de objetos a través de estos enfoques presenta algunos problemas que hasta la fecha son motivo de investigación: Uno de ellos consiste en la pérdida de información que se lleva a cabo al extraer la información geométrica. Un ejemplo es la pérdida de la información contenida dentro del contorno del objeto, la cual puede ser importante en muchas tareas de reconocimiento, este problema se ilustra en la Fig. 1.



Figura 1: (a) un objeto<sup>1</sup>. (b) posible forma de modelar el objeto en (a) a través de su contorno. Nótese que la información en forma de texto e imágenes grabadas sobre el objeto ya se ha perdido.

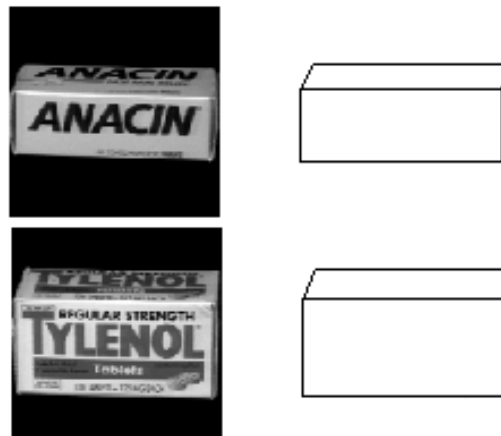


Figura 2: Dos objetos<sup>1</sup> (izquierda) y sus posibles modelos geométricos correspondientes (derecha), modelados a través de sus contornos.

<sup>1</sup> Imágenes tomadas de la base de imágenes COIL (Columbia Object Image Library), creada por el Dept. of Computer Science, Columbia University [Na96b].

Otro problema se encuentra al tratar de representar objetos que presentan una alta complejidad geométrica, como algunos objetos industriales, en donde se requiere una gran capacidad del sistema de reconocimiento.

Un problema mas, es la estabilidad en el cálculo de las características geométricas. Por ejemplo, un algoritmo para la extracción de arcos de contornos puede originar diferente número de arcos y diferente tamaño en dos corridas diferentes, sobretodo porque los parámetros usados para el algoritmo en el momento de la construcción del modelo, no son conocidos en el momento del análisis de la imagen donde se busca el objeto [AI96].

Un último problema que se citará aquí, se encuentra cuando se trata de describir objetos que poseen características geométricas semejantes, como por ejemplo, dos cajas de medicamentos, pues ambas se representarían mediante modelos geométricos extremadamente parecidos, lo cual implica tener ambigüedad en el proceso de reconocimiento (Fig. 2).

Analizando estos problemas, se puede observar que tienen un origen común: no es posible modelar completamente a un objeto solamente a través de sus propiedades geométricas, pues al hacerlo se pierde una cantidad muy significativa de información acerca de él, información que puede ser crucial cuando se le quiera reconocer. Aunado a esto, los enfoques geométricos usualmente tratan de representar al objeto de manera tridimensional, lo que requiere una gran capacidad de procesamiento del sistema de reconocimiento.

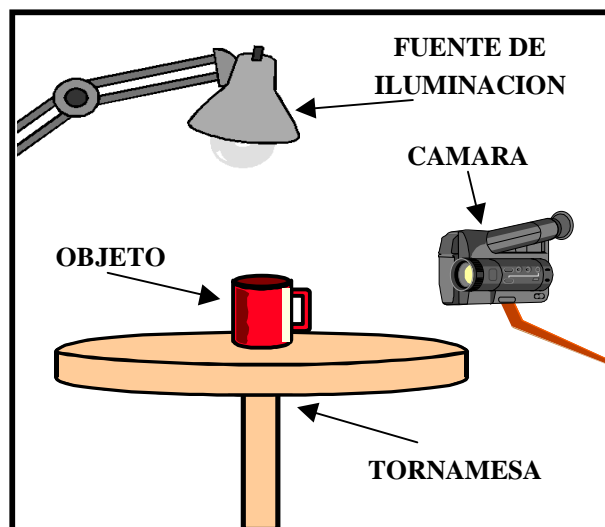


Figura 3: Sistema para la adquisición de imágenes. La fuente de iluminación y la cámara pueden tomar cualquier posición con respecto al objeto.

En la década de los 90's se ha propuesto el uso de *modelos basados en la apariencia* [Sh94] para representar a los objetos, de tal manera que es posible mediante el uso de ellos preservar no solamente las características geométricas de los objetos, sino también otro tipo de información tal como la textura, el color y otras características contenidas dentro del contorno.

Típicamente, este tipo de enfoque trabaja de la siguiente manera: se toma una cierta cantidad de imágenes del objeto del que se quiere construir el modelo, una para cada una de las posibles formas en que se piensa que el objeto podrá aparecer en una imagen, y combinado con las diversas condiciones de iluminación que incidan sobre él. En la Figura 3 se muestra un esquema de la forma en que se realiza la toma de imágenes. Enseguida se almacenan estas imágenes en algún formato elegido (gif, tiff, bmp, etc.), para su posterior uso; estas imágenes por sí mismas pueden formar el modelo del objeto, o bien, se les pueden extraer algunas características que se piensa describen al objeto de manera precisa, y entonces éstas constituirán al modelo. Finalmente, el proceso de reconocimiento propiamente dicho, consiste en comparar el modelo con una nueva imagen que probablemente contenga al objeto (por supuesto, este último proceso requiere algún procesamiento extra sobre la nueva imagen, tal como la segmentación del objeto de interés en la imagen, interpolación entre imágenes, etc.).

De la exposición previa, se puede observar que este enfoque presenta también algunos problemas:

- aunque en teoría se requiere considerar todas las posibles formas en que el objeto puede aparecer en una escena, en la práctica es imposible realizarlo, por la cantidad infinita de imágenes que se requerirían,
- por la misma razón, en la práctica es imposible considerar todas las posibles condiciones de iluminación que pueden incidir sobre el objeto<sup>2</sup>, además
- los modelos creados son sensibles a las condiciones de iluminación presentes en el momento de realizar la toma de las imágenes, por otro lado,
- aún considerando solamente algunas formas en que se espera que el objeto aparezca en una escena, y con condiciones de iluminación controladas, la cantidad de espacio requerido para el almacenamiento del modelo puede ser extremadamente grande, por último,
- en general, esta clase de enfoques requiere que el objeto aparezca segmentado, o sea fácilmente segmentable del resto de la escena,

---

<sup>2</sup> Sin embargo, algunos trabajos como [Ep96], [Be98a], [Be98b], [Ep94], sugieren que es posible obtener una representación compacta del objeto, bajo todas las posibles condiciones de iluminación que puedan incidir sobre el objeto, si se mantiene fija la posición de éste respecto al sensor.

por citar solo algunos. Además, algunos expertos en el área coinciden en que existen algunas desventajas en el uso de los modelos basados en apariencia [P&H96], algunas de las cuales se describen a continuación:

- Los métodos geométricos ponen énfasis en modelar analíticamente la relación entre las características del objeto y sus proyecciones; en contraste, esta relación es tradicionalmente capturada de una manera empírica por las técnicas basadas en apariencia. Sin embargo, el análisis estructural y analítico de la apariencia está comenzando a emerger: trabajos como los de Epstein, Yuille y Belhumeur [Ep96], Belhumeur, Peter y Kriegman [Be98] (ver Sección I.1) proporcionan una idea sobre la estructura del espacio de las apariencias fotométricas.
- Una de las limitaciones de las técnicas basadas en apariencia, es que son métodos globales y por tanto, requieren algoritmos poderosos de segmentación para delinear los objetos de interés en la imagen. Sin embargo, esta opinión está siendo cambiada por técnicas como las propuestas por Pope y Lowe [P&L96b], Pauli, Benkwitz y Sommer [Pa95] y Ohba y Ikeuchi [Oh96] (ver Sección I.2), las cuales usan también información local para reconocer los objetos.

A pesar de la existencia de estos problemas y aparentes desventajas, diversos estudios realizados tanto en humanos como en macacos (ver, por ejemplo, [Pa96]), sugieren que los sistemas biológicos emplean la visión basada en vistas (en donde la visión basada en apariencia queda incluida), y por lo tanto, sería razonable reproducir esta forma de visión en sistemas computarizados. De hecho, algunas propuestas (ver, por ejemplo, [Pa96] y [Po96]) ilustran que no solo es viable tal implementación, sino que además muestra un buen comportamiento en términos del reconocimiento de objetos. Por tanto, el enfoque basado en apariencia está cobrando importancia, como un complemento y/o alternativa eficiente a los tradicionales enfoques geométricos.

## I.1 Definición y proceso de reconocimiento

El término **reconocimiento de objetos tridimensionales basado en apariencia** fue propuesto por Shapiro y Costa [Sh94], aún cuando ya existían ideas previas que describían el concepto [Tu91], [Mu92], [Po94], pero sin formalizarlo como un enfoque nuevo en el reconocimiento de objetos.

Suponiendo que se ha tomado un conjunto de imágenes de entrenamiento de un objeto, Shapiro y Costa definen **el modelo basado en apariencia de un objeto**, como una descripción del objeto en términos de cualesquiera características que sean detectables en las imágenes del objeto (no solo geométricas), éstas incluyen los diferentes valores de intensidad, textura, color, etc. Una característica es detectable si hay un programa de computadora que puede extraer la característica de una imagen del objeto, a través de algún procedimiento bien definido. Los modelos basados en

apariencia pueden ser *modelos completos* del objeto, incluyendo todas las características que aparecen en cualquier vista de un objeto, o pueden ser *modelos de una clase de vistas* en los cuales un objeto es representado por un pequeño conjunto de vistas escogidas, cada una teniendo su propio conjunto distintivo de características.

Aunque los enfoques basados en apariencia para visión computacional pueden diferir substancialmente entre ellos, todos ellos son basados en la premisa de que las descripciones tridimensionales de los objetos no son requeridas explícitamente para poder reconocerlos [He95]. La idea fundamental de los enfoques basados en apariencia es que un conjunto de vistas de un objeto captura toda la información relevante, eliminando la necesidad de describir explícitamente la geometría tridimensional. Este es el aspecto más relevante que los hace distintos de los tradicionales enfoques geométricos. Ejemplos de sistemas geométricos que modela explícitamente la geometría de los objetos pueden ser encontrados en [Me92] y [Ho95]. Los enfoques basados en apariencia más radicales, actualmente no almacenan ninguna información tridimensional. Sin embargo, esto no impide la existencia de sistemas híbridos basados en apariencia y en geometría a la vez (Ver Sección I.2.3). En [He95] se comenta que existen dos principales argumentos en favor del uso de representaciones basadas en apariencia. El primero es que la información dada por el sólo contorno no es suficiente para el reconocimiento de objetos; los modelos de los objetos deben incluir información sobre la apariencia también. El segundo argumento es que una secuencia de imágenes captura la geometría tridimensional del objeto. Estos argumentos son apoyados por algunas evidencias interesantes de que el sistema visual humano usa una representación basada en vistas [Po94].

El proceso de reconocimiento usando modelos basados en apariencia, normalmente involucra realizar las tareas mostradas en la Figura 4. Esta representación gráfica es semejante a la descrita por Pope en [Po94]. Los bloques rectangulares representan los componentes del sistema, que en conjunto realizan el reconocimiento. La tarea del bloque ***Creación de modelos*** tiene como objetivo crear el modelo que representará al objeto, tomando como entrada el conjunto de imágenes del objeto que se quiere modelar. Mediante el procesamiento de las imágenes, se extraen de éstas algunas características que se espera encontrar en cualquier escena que contenga al objeto, y que revelarán la presencia de éste en la escena. El ***Procesamiento de la escena*** se refiere a la detección de los posibles objetos dentro de la imagen de interés, y a la posible segmentación de cada uno de ellos del resto de la escena, puesto que se requiere reconocer cada uno por separado. En esta etapa también se puede requerir la extracción de características que se espera formen a algún modelo, previamente almacenado en la base de modelos. La ***Comparación (matching)*** realiza el proceso de comparar las características obtenidas en la fase previa, contra las características que forman a los modelos en la base de modelos, seleccionando aquellos que presenten algún grado de semejanza con el objeto a reconocer, generando hipótesis acerca de la identidad del objeto en cuestión. Estas hipótesis son

almacenadas para ser empleadas en la etapa de *Verificación de hipótesis*, en donde se selecciona aquel modelo que mas se asemeje al objeto que se intenta reconocer, revelando de esta manera tanto la identidad de éste, como la localización dentro de la escena y los parámetros espaciales y de iluminación, desde donde se realizó la toma de la imagen que lo contiene (escena).

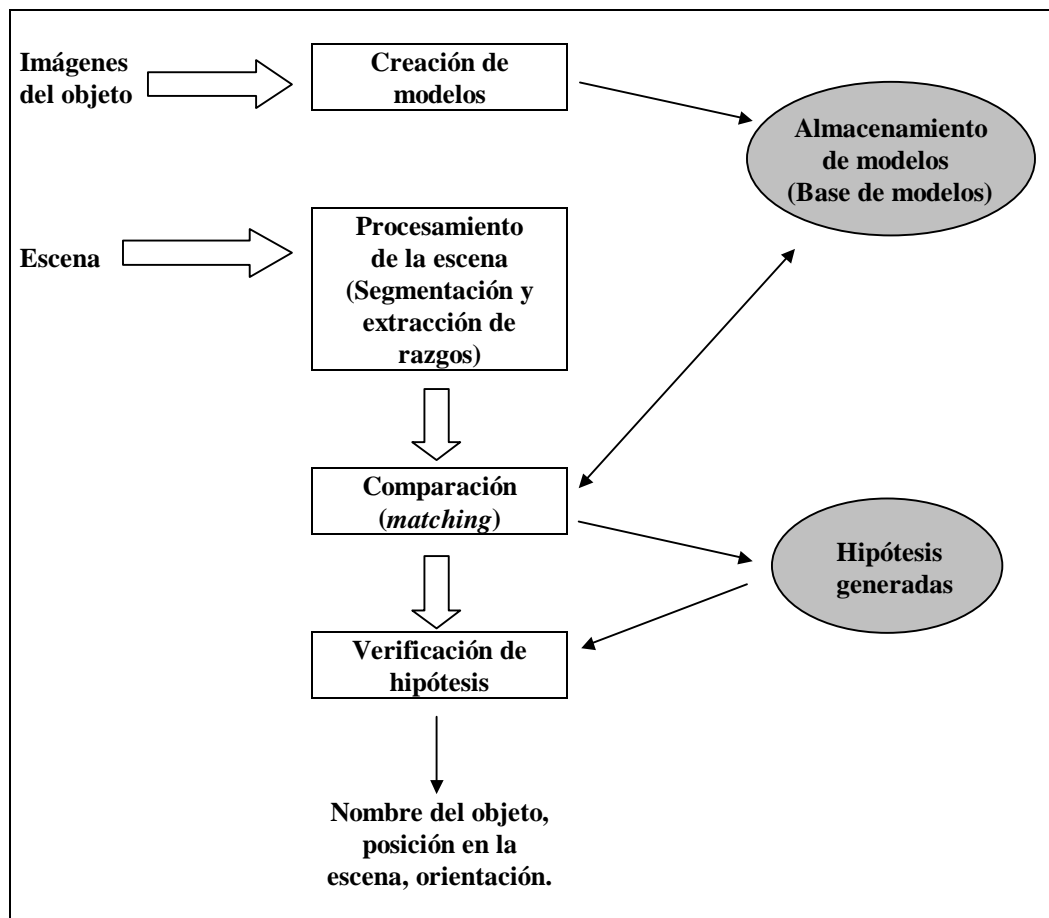


Figura 4: Componentes típicos de un sistema de reconocimiento que emplea modelos basados en apariencia.

Se puede observar que dependiendo de la forma en que se realicen las tareas descritas, se obtendrán distintos sistemas y de ello también dependerá la calidad de los resultados obtenidos. Tomando como base esta información, las Secciones I.2 a I.5 describen las ideas que diversos autores han propuesto, para realizar cada una de las 4 tareas descritas. Enseguida, en la Parte II del presente Capítulo se expone algunos resultados experimentales, comparando la eficacia del reconocimiento entre 2 sistemas geométricos y un sistema basado en apariencia.



## I.2 Creación de Modelos

Como se mencionó en la Sección anterior, la creación de modelos se refiere a la forma en que los objetos son representados y almacenados en la computadora, para su posterior uso en la etapa de reconocimiento. En general, cada propuesta difiere significativamente de las demás porque crea su propia representación, dependiendo de la tarea en que se usará el sistema de reconocimiento.

Shapiro y Costa [Sh94], por ejemplo, proponen la aplicación del operador detector de bordes Canny para obtener el contorno del objeto, y enseguida, aplicar filtros para detectar varias características que guiarán el reconocimiento, como por ejemplo, segmentos de línea recta, arcos circulares, elipses, etc., adicionando relaciones entre ellas, para crear una descripción estructural.

La propuesta anterior es similar a la que hace Nelson y Selinger [Ne97], [Ne98a], [Ne98b], [Ne98c], en cuanto al empleo solamente del contorno del objeto, sin embargo, ellos lo procesan en 4 niveles de agrupamiento perceptual: el primer nivel consiste en tomar una imagen del objeto y obtener los contornos de éste; el segundo nivel consiste en obtener características claves que guíen el posterior proceso de reconocimiento (una característica se llama característica clave, si con ella podemos identificar con un alto grado de certidumbre al objeto al que pertenece). Estas características deben ser lo suficientemente complejas como para especificar la configuración del objeto, además de tener una probabilidad sustancial de detección y ser insensibles a las poses tanto como sea posible. Estas características las aíslan en “parches”, que en conjunto representan al objeto. Un parche, por tanto, contiene una porción del contorno del objeto que cumple con las condiciones mencionadas. El tercer nivel de agrupamiento consiste en la formación de una vista del objeto, compuesta de los parches que forman al objeto en una imagen tomada desde algún punto de vista. Por último, el cuarto nivel de agrupamiento consiste en agrupar todas las vistas del objeto, para formar su modelo. Aún cuando el concepto de “parche” ya existía previamente, este enfoque difiere de los previos en que no requiere construir un modelo tridimensional del objeto, ni tampoco requiere almacenar su información tridimensional. Además, el concepto de parche se aplica a pequeñas secciones (claves) del contorno del objeto, no a una vista completa, como se aplicaba en enfoques previos.

Radicalmente distinta y poderosa es la propuesta de Murase y Nayar [Mu92], [Na95b], [Na96d], porque aún cuando no emplean todas las características del objeto, si capturan las características principales incluyendo por supuesto, el contorno. El método parte de capturar imágenes del objeto, tomadas desde distintos puntos de vista dentro de la esfera de visión, y procede a aplicarles el Análisis de Componentes Principales (PCA) [Oj83], basado en la transformación de Hotelling [Go96], para obtener una base ortonormal de vectores (imágenes propias ó *eigenimages*) que

permite representar cualquier imagen del conjunto como una combinación lineal de ellas. Proyectan entonces todas las imágenes a un espacio multidimensional para formar el modelo del objeto. Como es computacionalmente costoso calcular todos los vectores y valores característicos correspondientes, únicamente se calculan los más significativos, obteniendo solamente un subconjunto de ellos, con lo que se reduce la dimensión del espacio requerido para representar el objeto. Esta técnica es poderosa porque emplea una mayor cantidad de información acerca del objeto, que las que se basan solamente en la detección del contorno.

Aún cuando es poderosa la propuesta de Nayar y Murase, requiere que los objetos a reconocer no se encuentren parcialmente ocultos, porque de otra manera, el proceso de reconocimiento fallará. Para enfrentar este problema, Ohba e Ikeuchi [Oh96] proponen el uso del concepto de ventanas fundamentales (Eigen-Windows), que divide cada imagen que se usa para formar el modelo, en pequeñas “ventanas”, cada una conteniendo una parte de la imagen original. A todas esas ventanas se les aplica posteriormente el método de Murase y Nayar, proyectándolas finalmente al espacio fundamental. De esta manera, lo que se espera reconocer posteriormente son las ventanas en lugar de las imágenes completas. Desafortunadamente, el empleo de este método genera una gran cantidad de puntos en el espacio fundamental, por lo que probablemente requiera mayor cantidad de tiempo de cómputo. Continuando con estas ideas, Ohba, Sato e Ikeuchi en [Oh98], retoman el uso tanto de la técnica de Eigen-Windows y proponen alcanzar invariancia respecto a la iluminación, usando imágenes en color en formato RGB, en lugar de usar imágenes en tonos de gris como se supuso en [Oh96], transforman éstas en imágenes a formato HSV (H: *Hue*, S: *Saturation*, V: *Value*), adicionando un parámetro más denominado fase (*phase*).

Sobre estas mismas ideas en el uso de PCA, Nayar, Baker y Murase en [Na96e], proponen un mecanismo que detecta cualquier característica paramétrica. Estas ideas tienen fundamento en que tradicionalmente, cuando se desea detectar alguna característica de una imagen (por ejemplo, arcos, líneas, etc.), se debe construir un detector particular para cada tipo de característica. Aún cuando ellos mencionan algunos trabajos anteriores, exponen también que el presente enfoque difiere significativamente de los previos, en que aquí se asume que las características a detectar no están restringidas en cuanto a la apariencia que pueden presentar, esto es, se capturan (de una manera discreta) todas las posibles maneras en que puede aparecer la característica.

El enfoque parte de la parametrización de la característica a localizar, usando como parámetros las posibles maneras en que ésta puede aparecer en una imagen (apariencia de la característica) y obteniendo imágenes para cada uno de los valores que adoptan los parámetros. Enseguida, se propone proyectar estas imágenes a un espacio fundamental como en [Mu95], para comprimir las. Los puntos así proyectados forman el modelo de la característica. El proceso de detección propiamente dicho, se

aplica sobre una imagen en la cual se desea detectar la característica, verificando para cada pixel (junto con la vecindad que lo circunda), la existencia de la característica, a través de proyectar el pixel (junto con su vecindad) a un punto en el espacio fundamental y localizar el punto más cercano del modelo al punto que se está procesando, y verificando si la distancia entre ellos es pequeña. Si es así, se determina que la característica está presente, de otra forma, la característica no está presente. Es importante notar que a simple vista, el enfoque puede parecer computacionalmente caro, pues requiere muchas comparaciones para funcionar, pero usando las técnicas de búsqueda propuestas en [Ne95], [Ne96], y la técnica de rechazadores (rejectors) propuesta en [Ba95], [Ba96], el algoritmo es eficiente, como se puede observar de los resultados experimentales que ahí se reportan. Una exposición bastante más detallada del material se puede encontrar en [Na95a].

Otra propuesta es presentada por Pauli, Benkwitz y Sommer [Pa95]; parte de la toma de un conjunto de imágenes que muestran al objeto en diferentes poses, y bajo variadas condiciones de iluminación. En seguida, procede a aplicar un detector basado en ondeletas (*wavelets*), para transformar la imagen del objeto en una ondeleta, determinado por la respuesta del operador sobre la imagen del objeto. En seguida procede a trazar dos líneas perpendiculares sobre la imagen del objeto, cruzándose en el centro de ésta, que servirán para muestrear la estructura (respuesta del operador) en esas direcciones. Este muestreo genera dos tipos de muestras, uno en la dirección X y otro en la dirección Y. Este proceso se realiza para todas las imágenes del objeto, para obtener un conjunto de muestreos en la dirección X y otro para la dirección Y. Estos conjuntos servirán para la caracterización del objeto, sin embargo, se puede notar que cada uno contendrá tantos elementos como imágenes del objeto se hallan procesado de la manera anteriormente descrita; de aquí se puede observar que es deseable reducir esta cantidad para reconocer al objeto más eficientemente, por lo que se procede a agrupar los elementos de los conjuntos mediante un criterio de similitud, usando redes neuronales.

Pope y Lowe [P&L96a] modelan a los objetos por una distribución de probabilidades que describen el rango de variación posible en la apariencia del objeto. Esta distribución de probabilidades es organizada en dos niveles. Variaciones grandes son manejadas por particionar las imágenes de entrenamiento en grupos que corresponden a vistas diferentes o configuraciones del objeto. Dentro de cada grupo, pequeñas variaciones son representadas por distribuciones de probabilidad definiendo varias características en la apariencia. Para cada característica en general, los autores representan su probabilidad de detección en varias posiciones con respecto al objeto completo, y la probabilidad de la característica teniendo varios valores para un caso en particular. Ideas similares sobre el uso de la probabilidad para modelar la apariencia de los objetos, pueden encontrarse también en [P&L96b].

Un trabajo interesante por los resultados formales que ahí se discuten, es el realizado por Belhumer, Kriegman y David [Be98a]. Ellos plantean que si se tienen las imágenes de un objeto bajo todas las posibles condiciones de iluminación que pueden incidir sobre él, y estas imágenes están formadas por  $n$  píxeles, entonces si se consideran éstas como puntos en  $\mathbf{R}^n$ , el conjunto de todas ellas es un cono convexo en  $\mathbf{R}^n$  (cono de iluminación), que puede ser construido con tan pocas como 3 imágenes del objeto. Este resultado es relevante porque permite, para una pose fija, capturar la apariencia del objeto bajo todas las posibles condiciones de iluminación, con tan pocas como 3 imágenes del objeto. Este trabajo representa una generalización del trabajo de Epstein, Yuille y Belhumeur [Ep96], porque mientras que éstos suponen un modelo de iluminación Lambertiano, y en base a él capturan la apariencia del objeto, Belhumer, Kriegman y David en [Be98a], suponen condiciones más generales de iluminación. Estos resultados son apoyados por resultados empíricos como los obtenidos por Epstein, Hallinan y Yuille [Ep94], donde se muestra que  $5 \pm 2$  imágenes (*eigenimages*), son suficientes para capturar un porcentaje elevado (aproximadamente el 90%) de la variancia que presenta el objeto bajo condiciones de iluminación variable, para una pose fija. Como se puede observar, estos trabajos capturan la apariencia del objeto para condiciones de iluminación variable, pero además, tratan de resolver uno de los problemas citados en la introducción: reducir la cantidad de imágenes necesarias para modelar un objeto. Otras formas de resolver este mismo problema son discutidas en la siguiente Sección.

### **I.2.1 Compresión**

Dado que los sistemas que emplean modelos basados en apariencia requieren procesar una gran cantidad de imágenes, una parte importante para el funcionamiento de ellos, se refiere a la posibilidad de comprimir éstas de alguna manera, para poder reducir tanto el espacio de almacenamiento como el tiempo de procesamiento.

Nayar y Murase [Na96a], por ejemplo, proponen aplicar el Análisis de Componentes Principales (PCA) para comprimir las imágenes en un modelo compacto, empleando solo algunas de las imágenes fundamentales para construirlo. Pero, aplicar el método de Nayar y Murase requiere disponer de todas las imágenes a procesar sin estar previamente comprimidas. Sin embargo, debido a la gran cantidad de ellas, es deseable comprimirlas hasta el momento de aplicar PCA. El mayor problema se encuentra al descomprimirlas pues esto requiere de cómputo adicional. Para evitar esto, Seales y otros [Se96], proponen procesar las imágenes sin descomprimirlas y así salvar tanto espacio para su almacenamiento, como tiempo de procesamiento. Ellos muestran que es posible realizar el reconocimiento de objetos usando modelos basados en apariencia, almacenando las imágenes del objeto en los formatos MPEG ó JPEG, y sin descomprimir éstas, aplicar PCA para obtener las proyecciones en el espacio fundamental que corresponde al objeto. La propuesta se basa principalmente en la idea de que la Transformación Discreta del Coseno (DCT)

preserva la distancia en el espacio fundamental, permitiendo los mismos resultados en la clasificación sobre las imágenes transformadas, y dado que los estándares de compresión MPEG y JPEG están basados en DCT, entonces es posible realizar las operaciones necesarias para el reconocimiento de objetos, en el espacio de imágenes comprimidas. La parte más importante de este trabajo se refiere a la manera en que es posible realizar el cálculo de los vectores característicos y sus correspondientes valores característicos. Considerando que existe una pérdida de precisión al comprimir las imágenes, debido al proceso de cuantización, se muestra cómo los vectores característicos y valores característicos obtenidos realizando cálculos en el espacio comprimido, son una buena aproximación de los correspondientes calculados en el espacio no comprimido. Una extensión de este trabajo que usa esquemas de compresión con relaciones de compresión más elevadas se puede encontrar en [Hu98].

## **I.2.2 Interpolación**

Como se ha mencionado anteriormente, cuando se usan modelos basados en apariencia para realizar el reconocimiento de objetos, se requiere la realización de tomas que capturan las diversas vistas que puede presentar el objeto, considerando variaciones tanto en posición como en condiciones de iluminación. La tarea es tediosa y requiere una cantidad considerable de tiempo, aún cuando se realice de manera automática. Esto es cierto en particular para el método de Nayar y Murase, por lo que Seales y Yuan [Se98] proponen una técnica que se basa en la generación de tomas artificiales obtenidas a través de interpolación (*morphing*) entre tomas reales consecutivas, y que son utilizadas (junto con las demás tomas reales) como entrada para aplicar PCA. Aún cuando la técnica ya ha sido empleada previamente [Po94], no se había propuesto su aplicación para la propuesta de Murase y Nayar [Mu92]. El primer problema que plantean es que la técnica de reconocimiento de objetos usando PCA requiere una cantidad considerable de imágenes del objeto. Aunado a esto, surge el segundo problema: el uso de PCA puede fallar en el reconocimiento de objetos, debido a translaciones y cambios de escala de los objetos que se reconocerán. El primer problema es parcialmente resuelto al realizar tomas del objeto menos cercanas entre sí (aumentando el ángulo de separación entre tomas), y aplicando una técnica de interpolación (*morphing*) para obtener imágenes artificiales entre cada par de tomas reales consecutivas. Sin embargo, la efectividad de la técnica depende de las imágenes reales que se usen, por lo que una pregunta que surge es: ¿Cuáles y cuántas son las imágenes reales del objeto con que se obtendrán los mejores resultados? El segundo problema es resuelto observando que se puede obtener una gran cantidad de imágenes artificiales a distintas escalas y translaciones que permitirán mejorar el reconocimiento. Sin embargo, se debe observar que aumentar la cantidad de imágenes usadas con PCA incrementa el tiempo requerido para calcular los vectores característicos y valores característicos de la matriz de covarianza [Mu92], por lo que hace que la técnica no parezca aplicable para resolver este último problema.

Poggio y Beymer [Po96], proponen el uso de redes neuronales para el reconocimiento de objetos; en particular, muestran que no se requiere alimentar la red con una gran cantidad de vistas, pues se usa la ventaja de la interpolación que intrínsecamente la red realiza.

### **I.2.3 Modelos híbridos**

Aún cuando tradicionalmente, los enfoques geométrico y basados en apariencia se han considerado distantes unos de otros, algunos autores han propuesto sistemas que aprovechan las ventajas de ambos enfoques.

Ejemplos de este tipo de sistemas, son los propuestos por Burge y Burger [Bu97] que procesa cada una de las imágenes que capturan la apariencia del objeto, para describirlo en términos de sus partes y las relaciones entre ellas, permitiendo reconocer clases de objetos. Otro enfoque es el propuesto por Byne y Anderson [By98], que describe como es posible el uso combinado de la visión basada en modelos CAD y la visión basada en apariencia, para lograr una solución más general y robusta. Se sabe que ambos enfoques presentan limitaciones, como por ejemplo, al usar modelos CAD se puede manejar una estimación poco precisa de la textura del objeto, además de que la modelación de esta característica no es una tarea fácil. Por otro lado, los modelos basados en apariencia trabajan bien con los objetos que son difíciles de manejar usando métodos puramente geométricos, como por ejemplo, objetos con múltiples y variables texturas. Sin embargo, este tipo de modelos tienen algunos problemas con objetos semiocultos, y objetos cuyas distancias u orientaciones no están presentes en las imágenes que forman al modelo. Este enfoque parte de la admisión como entrada de un modelo CAD, tomado de (o construido con) algún paquete comercial, y lo transforma en un modelo que puede ser usado para reconocimiento. Esta conversión se realiza a través de construir “parches” planos del modelo CAD, uno para cada posible vista seleccionada, modelando al objeto a través de una aproximación poliédrica. Adicionalmente, a estos “parches” se les adiciona información extra que puede ser útil en el proceso de reconocimiento, como: orientación, tipo de superficie modelada, tipo de material que lo compone, etc. Una vez obtenido el modelo, se procede a realizar las tomas del objeto real, para cada una de las poses que componen al objeto, y se construyen vectores de características como color y textura, tomadas de estas imágenes.

## **I.3 Procesamiento de la escena**

Todos los enfoques mencionados aquí, generalmente requieren de un poderoso algoritmo de segmentación, para aislar a cada uno de los objetos del resto de la escena. Por supuesto que este es un punto de gran importancia para el reconocimiento de los objetos, pero la mayoría de los autores se enfoca en los procesos de creación de los modelos y la comparación de éstos, con los objetos encontrados y aislados de la

escena, suponiendo que es posible tal segmentación de una manera exitosa. Esto se debe a que el proceso de segmentación es considerado como un problema en sí mismo, extremadamente difícil de resolver en el caso general, y que permanece como un problema abierto. Sin embargo, de los resultados obtenidos en esta etapa depende en gran medida el éxito del sistema.

Existen algunas excepciones que no presuponen tal segmentación, como el enfoque que presentan Pauli, Benkwitz y Sommer [Pa95], que filtra la imagen mediante ondeletas (*wavelets*) y concentra su atención en aquellos píxeles (y sus respectivas vecindades) que presentan una alta respuesta al filtro, localizando de esta manera a los objetos en la escena.

Nelson y Selinger [Ne97], aplican un detector de bordes para localizar a los objetos. Posteriormente, segmenta las regiones localizadas para encontrar las características claves, que guiarán al proceso de reconocimiento.

Ohba e Ikeuchi [Oh96], dividen la escena en pequeñas ventanas que contienen porciones de los objetos, y las proyectan al espacio fundamental para posteriormente comparar estas proyecciones con los modelos de los objetos.

#### **I.4 Comparación (*matching*)**

La etapa de comparación, se refiere esencialmente al proceso de búsqueda dentro de la base de modelos del sistema, del modelo cuyas características se aproximen más a las localizadas en la imagen en que se desea reconocer un objeto. Esta etapa al igual que la previa (procesamiento de una escena) es dependiente del tipo y forma en que se han creado los modelos.

Para reconocer un objeto, el método de Nayar y Murase requiere proyectar la imagen que lo contiene a un punto en el espacio propio (eigenspace), para entonces encontrar el punto más cercano que pertenezca a algún modelo dentro del espacio propio. Debido a que un modelo queda definido por una gran cantidad de puntos, cuando se tiene almacenados una gran cantidad de modelos, esta búsqueda se vuelve una tarea crucial. Por esto, Nene y Nayar [Ne95], [Ne96], proponen un algoritmo de búsqueda en espacios multidimensionales, que permite, dado un punto en el espacio (vector), localizar, dentro un conjunto de puntos, el punto más cercano a él, de una manera eficiente.

Una versión simplificada de la propuesta de Nayar y Murase es expuesta por Pontil y Verri en [Po97]. Ahí se propone considerar a las imágenes de los objetos, como vectores (uniendo las columnas, una tras de otra) y con ellos crear un espacio que representa a los objetos, es decir, no aplican PCA para comprimir las imágenes; esto, por supuesto, es impráctico, pues requiere una gran capacidad de almacenamiento

del sistema. No obstante, algo que llama la atención de esta propuesta, es que ellos reducen el proceso de reconocimiento, a encontrar hiperplanos que separan a los modelos de los objetos, y cuando se quiere reconocer un objeto en una imagen nueva, simplemente se considera ésta como un vector (como se mencionó antes), y se determina de qué lado del hiperplano se encuentra, reconociéndolo como un el objeto que se encuentra de ese lado del hiperplano. Sin embargo, a pesar de la simplicidad del proceso, inmediatamente se puede observar que existirán problemas cuando los modelos se intersecten en el espacio (cuando los objetos son muy semejantes), requiriendo además una gran capacidad de almacenamiento del sistema. En el trabajo se reportan buenos resultados al reconocer nuevas imágenes de objetos.

El reconocimiento de objetos puede ser reducido en algunos casos, a un problema clásico de reconocimiento de patrones, como se puede observar en [Mu95]. Sin embargo, cuando el número de clases es grande, la eficiencia del reconocedor es un aspecto crítico que requiere atención especial. En consecuencia, Baker y Nayar [Ba95], [Ba96] proponen la noción de rechazador (*rejector*), que permite alcanzar una mejor eficiencia en tareas que involucren el reconocimiento de patrones, en particular, para el reconocimiento de objetos usando modelos basados en apariencia. Aunado a este concepto, proponen una teoría que amplía el concepto de rechazadores a rechazadores compuestos, y realizan un análisis de la eficiencia alcanzada al usar estos conceptos sobre el reconocimiento de patrones. Un rechazador se puede concebir como una generalización del concepto de reconocedor, en el sentido de que, mientras este último entrega como resultado una única clase  $W$  a la cual pertenece el elemento a reconocer, el rechazador entrega un conjunto de clases  $X$ , a una de las cuales pertenece el elemento. Usando la salida de un rechazador como entrada para otro(s) rechazador(es), - lo que dá lugar al concepto de rechazador compuesto -, es posible reducir en cada etapa el espacio de búsqueda (número de clases) y como consecuencia lograr una mejor eficiencia. Ellos aplican en particular su propuesta al método de Nayar y Murase [Mu95] reportando mejores resultados en términos de eficiencia y porcentaje de reconocimiento.

Nelson y Selinger [Ne97] almacenan los modelos en una base indexada por las claves (parches) que componen a los modelos, y proceden a verificar que modelos contienen a cada una de las claves que fueron localizadas en la escena a reconocer, almacenando las referencias a ellos para posteriormente verificar estas hipótesis.

Pauli, Benkwitz y Sommer [Pa95] proponen el uso de redes neuronales para realizar la comparación. Es interesante mencionar que con esta propuesta, no se requiere segmentar la escena a reconocer, pero se requiere localizar puntos con características particulares, que definen porciones de la escena, que posteriormente serán procesadas por las redes previamente entrenadas. Poggio y Beymer [Po96], paralelamente, proponen también el uso de redes neuronales para el reconocimiento de objetos; en particular, muestran que no se requiere alimentar la red con una gran



cantidad de vistas, pues se obtiene ventaja de la interpolación que intrínsecamente la red realiza.

Un punto importante que se debe notar, es que muy pocos enfoques toman en cuenta las condiciones de iluminación que inciden sobre los objetos. En general, se asumen condiciones de iluminación controladas. Si bien esta suposición es razonable para varias aplicaciones, se sabe que el humano puede reconocer objetos en condiciones no controladas y aún en condiciones que tradicionalmente se han considerado difíciles de abordar en visión computarizada, como por ejemplo, cuando aparecen sombras de los propios objetos en la escena a reconocer, que se mezclan entre ellas mismas y entre los objetos, o bien, de objetos fuera de ésta, como también bajo condiciones de iluminación natural (luz solar), que hace que los objetos cambien de apariencia a medida que cambia la posición de la fuente de iluminación (sol).

Generalmente, se le ha dado gran importancia al modelado de la apariencia del objeto, manteniendo controladas las condiciones de iluminación, y por tanto, dejando un poco a un lado el estudio de la influencia de la iluminación sobre la apariencia del objeto. Sin embargo, para poder construir un sistema de reconocimiento poderoso, se deberán considerar y por tanto modelar el aspecto de la iluminación. Belhumer, Kriegman y David [Be98a] han notado esto, que en particular es cierto si se usa apariencia para modelar los objetos; por esta razón, Belhumer y Jacobs [Be98b] abordan el problema de determinar cuando dos imágenes provienen de un mismo objeto, pero bajo diferentes condiciones de iluminación. Ellos demuestran para algunos tipos de objetos relativamente simples, es posible determinar si dos imágenes pertenecen al mismo objeto, pero bajo diferentes condiciones de iluminación, o bien, que pertenecen a distintos objetos. Esta propuesta es aplicada al reconocimiento de rostros, mostrando que se obtienen mejores índices de reconocimiento que por el uso de correlación o PCA.

## **1.5 Verificación de hipótesis**

El último paso en el proceso de reconocimiento, es decidir si una hipótesis encontrada en el paso anterior es una instancia de un objeto en la imagen. Aquí hay relativamente poca investigación sobre como realizar el proceso de decisión [Po94]. Algunos sistemas simplemente requieren que alguna fracción (umbral) seleccionado empíricamente, de las características del modelo se encuentren en la hipótesis, y/o viceversa. Este es el caso de los enfoques de Shapiro y Costa [Sh94] y de Nelson y Selinger [En97], sin embargo, se puede observar que esta idea no proporciona los mejores resultados, debido a que solamente se considera la cantidad y no la calidad de las características empleadas.

Algunos estudios se han realizado para considerar tanto la cantidad y la calidad de las características empleadas; un ejemplo es la propuesta presentada por Pope y

Lowe [P&L96a], en donde se emplea la Teoría Bayesiana para estimar la probabilidad de que una hipótesis particular, represente una ocurrencia verdadera del objeto en la imagen.

Finalmente, algunos enfoques no requieren el uso de esta etapa de verificación, debido a la naturaleza de su definición. Un ejemplo de este tipo corresponde a la propuesta de Murase y Nayar [Mu92], en donde se emplea la SSD (*Sum of Squared-Difference*) entre el punto que corresponde a la imagen del objeto a reconocer, proyectada en el espacio propio, al modelo más cercano a éste como medida de similitud, y dado que es difícil que esta distancia sea igual para dos modelos distintos, no se contempla esta posibilidad.

## **PARTE II: Enfoque geométrico vs. basado en apariencia**

### **II.1 Enfoques geométricos**

Los enfoques previos a la aparición de los enfoques basados en la apariencia, usualmente describen a los objetos en términos de su contorno, y no manejan propiedades adicionales como color y textura. A este tipo de enfoques se les conoce como *enfoques geométricos*. Por convención, usaremos el concepto de contorno tal y como lo sugiere Pope en [Po94]: “Contorno significará la geometría de un lugar geométrico de puntos, los cuales serán típicamente los puntos sobre la superficie de un objeto físico, o los puntos sobre la intensidad de un lado o región en una imagen”.

Por representación del contorno se entenderá, también por convención, el concepto señalado por Pope en [Po94]: “Una representación del contorno es un lenguaje para describir el contorno o algunos aspectos del contorno”. Las representaciones del contorno de los objetos que manejan los sistemas de reconocimiento basados en enfoques geométricos son muy variadas. Sin embargo, en general, todas ellas pueden ser agrupadas en términos de la dimensión del modelo que emplean:

- Algunos enfoques emplean una representación centrada en el objeto, lo que significa que requieren representar al objeto mediante un modelo tridimensional. Un ejemplo de un sistema de este tipo puede ser encontrado en [Me92].
- Otros enfoques manejan una representación basada en vistas o multivistas, lo que significa que construyen la representación del objeto mediante el contorno localizado en tales vistas (imágenes) del objeto, sin requerir una descripción tridimensional de éste. Un ejemplo de un sistema de este tipo puede ser encontrado en [Ne97].

Un aspecto importante a señalar, es que como ejemplo de un sistema geométrico multivistas, se acaba de señalar el sistema propuesto por Nelson y Selinger

en [Ne97]; sin embargo en la Sección I.2 se le citó como un enfoque basado en apariencia. Esto significa que tal sistema es a la vez un enfoque geométrico y también un enfoque basado en apariencia. Esta apreciación es correcta, pues es permitida por la definición de modelo basado en apariencia presentado en la Sección I.1, porque en tal definición se establece que un modelo basado en apariencia está formado por algunas características atribuibles al objeto, dentro de las cuales se encuentra el contorno. Esto implica que si sólo se emplea el contorno como característica para construir los modelos, entonces éstos estarán basados en la apariencia de los objetos. Pero además, por el concepto expuesto previamente sobre modelos geométricos multivistas, tal sistema será también de este tipo. Es en este sentido que los modelos basados en apariencia se consideran una generalización de los modelos basados en geometría. Sin embargo, los enfoques basados en apariencia más radicales, actualmente no almacenan ninguna información tridimensional, y no están basados solamente en el uso del contorno de los objetos.

Un último aspecto que se señalará aquí sobre modelos geométricos, es que aunada a la representación del contorno, también muchos de estos enfoques emplean propiedades que poseen los objetos, que son usualmente derivadas del contorno: el área, el perímetro, el número de agujeros que presenta el objeto, etc. Aunado a esto, el uso de invariantes y semi-invariantes, que son propiedades del objeto que no cambian (o cambia poco, respectivamente) si cambia la posición del objeto o las condiciones de iluminación, es también un campo activo de investigación y de creciente interés.

Para una análisis crítico detallado sobre las diversas propuestas que se pueden encontrar en la literatura sobre modelos geométricos, el lector puede consultar el trabajo de Pope en [Po94]. Para una exposición de las técnicas más comunes para la construcción de modelos geométricos, el lector puede consultar, por ejemplo [Sh87] y [Ho87].

## **II.2 Comparación entre enfoques geométricos y basados en apariencia**

Aún cuando en la literatura no abundan las comparaciones entre ambos enfoques, estas son importantes para poder formarse una idea, acerca de las ventajas y desventajas de los enfoques basados en apariencia en relación con los geométricos, por lo que aquí se cita brevemente los resultados obtenidos por Mundy y otros en [Mu96], al comparar tres sistemas para el reconocimiento de objetos: SLAM que es un sistema basado en apariencia, propuesto y desarrollado por Murase y Nayar, y descrito en detalle en [Ne94a] y [Ne94b]. Las ideas para el desarrollo de este sistema fueron mencionadas antes y son descritas en [Mu92], [Na95b], [Na96d]. Los otros dos sistemas son LEWIS y MORSE, ambos basados en ideas geométricas y descritos en [Mu96].

Las conclusiones que se desprenden de este trabajo experimental, se pueden resumir en los siguientes puntos, que se refieren sólo a los 3 sistemas investigados:

- Ambos enfoques tienen ventajas y desventajas que se complementan entre sí.
- Los modelos basados en apariencia, tienen la gran ventaja de no requerir una descripción formal para cada tipo de objeto, esto es, no requieren analizar el tipo de objeto que se procesa, pues el procedimiento es general para cualquier objeto.
- No es claro hasta el momento, como un modelo basado en apariencia puede ser generalizado, para incluir objetos que pueden ser considerados dentro de una misma clase.
- SLAM presenta la desventaja de requerir algoritmos poderosos para segmentar los objetos en una escena, y no puede manejar objetos semiocultos. Sin embargo, posteriormente se han propuesto ideas para resolver este problema, como por ejemplo, ver [Oh96].
- Los modelos basados en apariencia son sensibles a las condiciones de iluminación, y aunque SLAM normaliza las imágenes tanto en escala como en brillantez, en la presencia de sombras e iluminación variable, esta normalización es difícil de alcanzar. Sobre este punto, es de esperarse que la descripción geométrica del contorno, sea más invariante que normalizar los patrones de intensidad.



Figura 5: Conjunto de objetos sobre los que se ha experimentado SLAM: Software Library for Appearance Matching<sup>3</sup>.

<sup>3</sup> Imágenes tomadas de la base de imágenes COIL (Columbia Object Image Library), creada por el Dept. of Computer Science, Columbia University [Na96b].

Un aspecto importante que se debe resaltar, es que en este estudio experimental, solamente se utilizaron objetos simples, que es posible describir geoméricamente. Si se utilizaran objetos que no cumplieran esta propiedad, tanto LEWIS como MORSE fallarían, a diferencia de SLAM que, no importando el tipo de objeto, es capaz de crear su modelo, y si la segmentación es eficiente, es capaz de reconocerlo, en general. Más aún, si el tiempo que se requiere para construir el modelo (que solamente se realiza una vez) no es tomado en cuenta, SLAM es capaz de reconocer objetos complejos en tiempo real. Esta y otras conclusiones han sido obtenidas, para una base de 100 objetos complejos, reportadas en [Na94b] y [Na96b]. Estos objetos son mostrados en la Figura 5.

### **PARTE III: Aplicaciones de los modelos basados en apariencia**

El reconocimiento de objetos tiene diversas aplicaciones, sobre todo en aquellas actividades en donde es factible el reemplazo del ser humano, quizá debido a que son actividades peligrosas para él (aplicaciones industriales que requieren el manejo de maquinaria ó sustancias peligrosas), o porque son tareas que son factibles de automatizarse (uso de robots), logrando alcanzar rendimientos superiores al del humano, en términos de rapidez y precisión.

En esta Sección, se mencionan algunas de las aplicaciones que se citan en la literatura, y que incorporan el reconocimiento de objetos usando modelos basados en apariencia. Si bien estas aplicaciones se plantean de manera experimental y no representan una listado exhaustivo, si pretenden servir para dar un panorama general de la importancia potencial que tiene el reconocimiento de objetos en general, y el uso de modelos basados en apariencia, en particular.

#### **III.1 Posicionamiento de robots**

Nayar, Murase y Nene describen en [Na94a] (Ver también [Na95c] y [Na96c]), la aplicación de la propuesta hecha en [Mu92], [Na95b] y [Na96d], para el posicionamiento de un robot sobre un objeto, que previamente ha sido modelado usando espacios propios. Más explícitamente, se realizan las tomas del objeto, muestrándolo detalladamente para así capturar su apariencia. Enseguida, se comprimen estas imágenes usando PCA y finalmente se proyectan al espacio fundamental para obtener el modelo del objeto. En la fase de posicionamiento, se coloca el robot frente al objeto en una posición arbitraria, y se selecciona una posición determinada en donde se quiere que se detenga el robot; éste procede a realizar una toma del objeto en esa posición y la proyecta al espacio propio, localizando entonces la posición en que se encuentra actualmente. Esta información la emplea para calcular

los desplazamientos que requerirá para llegar a la posición final, y por último, se mueve hacia esa posición y se detiene.

En particular, uno de los experimentos que se reportan en este trabajo, se refiere a posicionar al robot sobre una tarjeta de circuito impreso, comparando la posición a la que llega el robot con la posición deseada, concluyendo que la precisión es suficiente como para permitir al robot llevar a cabo la tarea de insertar un circuito integrado en la tarjeta de circuito impreso. Por supuesto, esta precisión puede ser incrementada tanto como se desee, mejorando la precisión del muestreo del objeto. La técnica es factible de emplearse en otras tareas que requieran el posicionamiento de un robot, sobre un objeto complejo. Obsérvese que una gran ventaja de esta aplicación radica en que el modelado de la tarjeta de circuito impreso se realiza de una manera extremadamente simple y automática, no importando que tan compleja sea, y por tanto, no requiere intervención alguna del operador del sistema.

### **III.2 Reconocimiento de ademanes**

El uso de ademanes es algo muy común en el proceso de comunicación entre las personas. Este lenguaje es muy rico en cuanto a capacidad de expresión. Además, a través del análisis del ademán que una persona realiza, es posible determinar lo que la persona está haciendo, como por ejemplo, tocar una guitarra. Por supuesto, el análisis a través de una computadora puede hacerse usando un conjunto de sensores conectados a ésta. Sin embargo, realizar el análisis por medio de un proceso únicamente visual tiene varias ventajas: entre otras, el realizarlo así le permite al usuario una total libertad, evitándole molestias.

En particular, el uso de ademanes hechos solamente con los dedos de las manos representan un medio de expresión muy rico y versátil, por lo que no es extraño pensar en su uso para controlar la computadora o bien para comunicarle información. Actualmente, la inserción de ordenes a una computadora se realiza a través del teclado o del ratón, sin embargo, el uso de ademanes puede ser más versátil porque le permite una gran libertad al usuario. Por esto, Martin y Crowley en [Ma97], proponen una técnica para permitirle a la computadora reconocer este tipo de ademanes de manera experimental; el tipo de ademanes analizados es semejante a los utilizados por los sordomudos como medio de expresión. Los modelos que representan alguna señal hecha con los dedos de la mano son basados en apariencia, porque como ellos mencionan, mantener un modelo tridimensional de la misma es difícil, pues involucra tanto como manejar 27 grados de libertad. Además al realizar el proceso de comparación de una nueva imagen que contiene un gesto manual a reconocer, se requiere comparar esa imagen 2D contra el modelo en 3D, lo cual es computacionalmente caro.

El sistema mencionado anteriormente construye los modelos para cada ademán, a través de capturar un conjunto de imágenes que representan las posibles maneras en que una mano puede expresar la misma señal (esto es, con los dedos más separados ó más juntos, ó doblados). A estas imágenes se les aplica la técnica de Análisis de Componentes Principales (PCA), basada en la Transformación Hotelling [Go96], para proyectarlas a un espacio propio que representará al ademán a reconocer (los ademanes válidos para este trabajo experimental solamente son 6: uno es la mano cerrada y los otros son formados con un dedo extendido, con dos, etc.). El sistema de reconocimiento parte de la toma de imágenes de una mano que continuamente se encuentra cambiando de señal, y para cada una de éstas imágenes localiza la mano, la aísla, y la proyecta al espacio fundamental donde a través de una máquina de estados finitos va comparando este punto contra los que integran el modelo, distinguiendo (a través de esta máquina) entre estados finales y estados transitorios (los estados finales representan ademanes conocidos).

Iwai, Hata y Yachida [Iw98], proponen también el reconocimiento de ademanes al igual que en la propuesta anterior, pero reconociendo ademanes hechos con los brazos (como tocar un instrumento musical). Ellos proponen realizar esto, a través de detectar, - a partir de una secuencia de imágenes tomadas del ademán que una persona realiza -, la trayectoria que la mano sigue a través de la secuencia, aislando ésta en cada imagen, para posteriormente usar la transformación Karhunen-Lóeve (KL) para proyectar la trayectoria del ademán en un espacio propio, y construir de esta manera el modelo del ademán. El proceso de reconocimiento de un ademán desconocido se realiza a través de aplicar el anterior proceso a una secuencia de imágenes que representen al nuevo ademán, y comparar esta trayectoria descrita en el espacio fundamental con cada uno de los modelos previamente almacenados. Este último paso se realiza a través del uso del Modelo de Markov Oculto (*Hidden Markov Model*), aplicado a cada modelo y a la nueva trayectoria. Nótese que ellos no comparan directamente las imágenes que forman a los ademanes, sino calculan las trayectorias que ellos describen, y son éstas las que emplean en el reconocimiento. Esta comparación de trayectorias representa una diferencia respecto a las propuestas discutidas anteriormente.

### **III.3 Reconocimiento de rostros**

Una de las aplicaciones más estudiadas del reconocimiento de objetos, es el reconocimiento de rostros. Esto quizá se deba a la importancia social que esta tarea conlleva, pues cotidianamente el humano lo realiza de manera automática y efectiva, al relacionarse e interactuar con sus semejantes. Sin embargo, tal tarea no es fácil de realizar automáticamente por la computadora, debido a diversos factores que están involucrados, y que, en general, no presentan los objetos inanimados: los rostros cambian de acuerdo a las expresiones faciales, de tal manera que el rostro de una persona cambia dramáticamente para la computadora, dependiendo de la expresión

facial que muestre en un momento dado: sonriendo, llorando, estando molesto, etc.; también, los rostros cambian si la persona usa anteojos ó no, si se corta el pelo, el bigote ó la barba, etc., e inclusive, cambian conforme avanza la edad de la persona.

A lo largo de la historia de la visión computacional, diversos autores se han ocupado de este problema. Sin embargo, la mayor parte se ha enfocado solamente en los aspectos geométricos, que como se mencionó en la introducción, tienen algunas limitaciones. En consecuencia, algunos autores han propuesto el uso de los modelos basados en apariencia en el reconocimiento de rostros.

Quizá una de las primeras propuestas en este sentido se deba a Turk y Pentland [Tu91], que proponen realizar el reconocimiento de rostros, capturando la apariencia del rostro de la persona, en un conjunto de imágenes que muestran al rostro en diversas condiciones de iluminación. Enseguida, aplican PCA para comprimir las imágenes y obtener una representación compacta en un espacio propio. El proceso de reconocimiento parte de proyectar una nueva imagen de un rostro desconocido en el espacio propio, y comparar esta proyección con las de los modelos previamente creados, conocidos hoy como *eigenfaces*. Este enfoque trabaja bien si se restringe en extremo la apariencia del rostro, pero si se desea un modelo preciso, es impráctico e incluso imposible de aplicar, pues no es posible considerar todas las condiciones de iluminación, en que pueden aparecer los rostros. Para intentar lograr un reconocimiento más preciso y más general, Pentland, Moghaddam y Starner en [Pe94], proponen el uso de vistas, para generalizar el enfoque de eigenfaces, permitiendo el reconocimiento de rostros desde diversas poses. Además, proponen el uso del concepto de espacios fundamentales modulares, que consiste en almacenar en un espacio fundamental los rostros íntegros, y en otros las características particulares (nariz, boca, ojos) relevantes de un rostro, para poder refinar el proceso de reconocimiento, enfocándolo hacia estas características particulares. Estas ideas son discutidas más ampliamente en términos de técnicas bayesianas en [Mo96].

Por otro lado, Belhumeur, Hespanha y Kriegman en [Be97], señalan otro problema asociado al uso de eigenfaces: el uso de PCA crea una representación compacta de los rostros, pero tiende a fusionarlos en clases que no están bien delineadas en el espacio fundamental, de tal manera que el reconocimiento de rostros por este medio, presentará en algunos casos ambigüedades. Por esto ellos proponen la aplicación del Discriminante Lineal de Fisher, a la representación obtenida mediante PCA, para lograr separar adecuadamente los rostros proyectados en el espacio fundamental. Paralelamente, Gong, Ong y Loft [Go98], proponen el uso de redes neuronales para resolver el mismo problema. Moghaddam, Wahid y Pentland [Mo98], lo abordan también, proponiendo distinguir entre variaciones intrapersonales (variaciones debidas a cambios en la iluminación ó expresiones faciales de una persona) y variaciones extrapersonales (debidas a diferencias en la identidad de



diferentes personas), mediante la sustitución de los criterios de comparación usados en [Tu91], por criterios basados en un análisis Bayesiano.

Como se puede observar de la exposición previa, el uso de PCA es una práctica común para diversos autores. Sin embargo, una desventaja adicional que presenta y a la que normalmente se le presta poca atención, radica en que una buena parte de las variaciones entre vistas, es debida a las condiciones de iluminación. En consecuencia, como PCA no modela explícitamente éstas, requiere una gran cantidad de imágenes. Para reducirla, Georghiades, Kriegman y Belhumer proponen en [Ge98], aplicar las ideas presentadas en [Be98a] al reconocimiento de rostros, permitiendo reducir la cantidad de imágenes necesarias.

## **PARTE IV: PROBLEMÁTICA A RESOLVER**

Uno de los problemas relacionado con el reconocimiento de objetos mediante el empleo de modelos basados en apariencia, y que fue mencionado al inicio del presente Capítulo, radica en la cantidad de imágenes requeridas para construir los modelos de los objetos a reconocer. Esta cantidad puede ser grande si se requiere reconocer al objeto desde diversos ángulos y en distintas condiciones de iluminación. Por tanto, es deseable emplear la menor cantidad de imágenes posible, para evitar que los sistemas de reconocimiento requieran una gran cantidad de espacio de almacenamiento y tiempo de cómputo.

Algunos autores han considerado este hecho y han enfocado sus esfuerzos en esta dirección. Uno de estos trabajos, es el realizado por Belhumer, Kriegman y David [Be98a], quienes plantean que si se tienen las imágenes de un objeto bajo todas las posibles condiciones de iluminación que pueden incidir sobre él, y estas imágenes están formadas por  $n$  píxeles, entonces si se consideran éstas como puntos en  $\mathbf{R}^n$ , el conjunto de todas ellas describe un cono convexo en  $\mathbf{R}^n$  (cono de iluminación), que puede ser construido con tan pocas como 3 imágenes del objeto (para algunos tipos de objetos). Este resultado es relevante porque permite, para una posición fija del objeto frente al sensor, capturar la apariencia del objeto bajo todas las posibles condiciones de iluminación, con tan pocas como 3 imágenes del objeto (para algunos tipos de objetos). Sin embargo, la determinación de estas imágenes características requiere un cálculo intenso sobre un conjunto grande de vistas del objeto. Este trabajo representa una generalización del trabajo de Epstein, Yuille y Belhumeur [Ep96], porque mientras que éstos suponen un modelo de iluminación Lambertiano, y en base a él capturan la apariencia del objeto, Belhumer, Kriegman y David en [Be98a], suponen condiciones más generales de iluminación. Estos resultados son apoyados por resultados empíricos como los obtenidos por Epstein, Hallinan y Yuille [Ep94], donde se señala que  $5 \pm 2$  imágenes (*eigenimages*), son suficientes para capturar un porcentaje elevado (aproximadamente el 90%) de la variancia que presenta el objeto bajo condiciones de

iluminación variable, para una pose fija. Como se puede observar, estos trabajos capturan la apariencia del objeto para condiciones de iluminación variable, pero además, tratan de resolver uno de los problemas citados en la introducción: reducir la cantidad de imágenes necesarias para modelar un objeto. Sin embargo, si la posición del objeto respecto al sensor es variable, el problema prevalece.

Poggio y Beymer [Po96], proponen el uso de redes neuronales para el reconocimiento de objetos; en particular, señalan que no se requiere alimentar la red con una gran cantidad de vistas, pues se usa la ventaja de la interpolación que intrínsecamente la red realiza. Sin embargo, ellos no realizan ninguna propuesta para determinar cuál sería el conjunto de vistas de entrenamiento adecuado, para obtener buenos resultados, en especial, para objetos muy parecidos entre sí.

Seales y Yuan [Se98] proponen una técnica que se basa en la generación de tomas artificiales obtenidas a través de interpolación (*morphing*) entre tomas reales consecutivas, y que son utilizadas (junto con las demás tomas reales) como entrada para aplicar PCA. Aún cuando la técnica ya ha sido empleada previamente [Po94], no se había propuesto su aplicación para la propuesta de Murase y Nayar [Mu92]. Un problema que plantean es que la técnica de reconocimiento de objetos usando PCA requiere una cantidad considerable de imágenes del objeto. Este problema es parcialmente resuelto al realizar tomas del objeto menos cercanas entre sí (aumentando el ángulo de separación entre tomas), y aplicando una técnica de interpolación (*morphing*) para obtener imágenes artificiales entre cada par de tomas reales consecutivas. Sin embargo, la efectividad de esta técnica, al igual que la propuesta por Poggio y Beymer [Po96] mencionada antes, depende de las imágenes reales que se usen, por lo que una pregunta que surge es: ¿Cuáles y cuántas son las imágenes reales del objeto con que se obtendrán los mejores resultados?

Esta pregunta ha sido parcialmente abordada más recientemente por Cootes y otros en [Co00], en donde se ha propuesto el empleo de puntos de control sobre el contorno de rostros para determinar qué vistas son estrictamente necesarias para modelar cada rostro en particular, determinando cómo deben comportarse esos puntos a través de transformaciones de rotación del rostro, logrando una importante reducción en la cantidad de imágenes requeridas, y obteniendo un buen porcentaje de reconocimiento. Sin embargo, el ámbito de aplicabilidad de esta propuesta es restringido únicamente a rostros.

Mokhtarian y Abbasi [Mo00] emplean también puntos de control, pero sobre objetos más generales, y logran también una reducción importante en el número de imágenes requeridas, sin embargo, su propuesta es restringida al empleo únicamente del contorno de los objetos y a objetos que pueden ser descritos fácilmente mediante su geometría. Más aún, esta propuesta es incapaz de trabajar con el resto de la

información proveniente de la apariencia del objeto, como la textura, y que puede ser extremadamente importante en diversas aplicaciones.

Otros autores han tratado de reducir el tiempo de cómputo (más no el espacio de almacenamiento) requerido para construir los modelos de los objetos, para la técnica de espacios propios de Murase y Nayar [Mu92], mediante el empleo de algoritmos más rápidos para calcular los vectores y valores propios de la matriz de covarianza del conjunto de imágenes de entrenamiento. Este es el caso de la propuesta hecha recientemente por Chang y otros en [Ch00], en donde se emplea la teoría de matrices de rotación, para calcular el espacio propio en un tiempo menor que el requerido por otras técnicas como el método poderoso y el gradiente conjugado. Hall Marshall y Martín en [Ha00] proponen también una técnica veloz para recalcular el espacio propio previamente calculado de un conjunto de imágenes, cuando se le adiciona a este conjunto una o más imágenes nuevas de entrenamiento, como por ejemplo, cuando se adiciona un objeto más al sistema de reconocimiento.

Sin embargo, es bien sabido de la teoría de la complejidad de algoritmos, que la eficiencia de estos métodos es seriamente afectado por la cantidad de imágenes que tienen que procesar, ya que todos ellos requieren procesar la matriz de covarianza, el tamaño de la cual está determinado por la cantidad de imágenes de entrenamiento, por lo que sigue siendo altamente deseable reducir esta cantidad de imágenes que dichos algoritmos deben procesar, como un paso previo para obtener una mayor eficiencia.

Como se puede observar, la reducción de la cantidad de imágenes es un aspecto importante a considerar al emplear modelos basados en apariencia, por lo que justamente el objetivo de la presente tesis es plantear y tratar de resolver la pregunta:

¿ Cuáles y cuántas son las imágenes estrictamente necesarias de un objeto cualquiera, para reconocerlo mediante modelos basados en apariencia ?

Si bien el presente trabajo trata de resolver el problema planteado en la pregunta anterior, éste no debe ser confundido con otro problema muy relacionado y que pertenece al área de síntesis de vistas, que consiste en determinar cuántas vistas características de un objeto son suficientes para representarlo y reconocerlo. En efecto, y como es bien señalado por Epstein, Yuille y Belhumeur [Ep96], existen similitudes entre ambos problemas, como también varias diferencias. Quizá la diferencia principal radique en que en síntesis de vistas se pretende determinar un conjunto de vistas del objeto, a través del cual, y mediante la combinación de sus elementos de alguna forma, se obtenga una representación del lugar geométrico determinado por todas las posibles formas en que puede aparecer un objeto en una imagen, considerando todas las posiciones que éste puede adoptar frente al sensor, y bajo cualesquiera condiciones de iluminación, en general. En contraste, en modelos basados en apariencia se trata de aproximar una trayectoria que la apariencia del

objeto describe dentro de dicho lugar geométrico, para condiciones específicas de iluminación y posición.

Más aún, la presente tesis no trata de resolver el problema antes citado de síntesis de vistas, ni siquiera para condiciones específicas de iluminación y de posición. Este problema es aún tema de investigación activa. En efecto, esta tesis únicamente pretende obtener una reducción significativa en la cantidad de imágenes requeridas, para representar paramétricamente la apariencia de un objeto cualquiera, cuando éste gira frente al sensor por un cierto ángulo  $\theta$ , respecto a una posición inicial fija, y bajo condiciones específicas de iluminación. Si la pregunta hecha anteriormente es planteada como un problema de optimización, en donde determinar la cantidad de imágenes requeridas es la variable de optimización del problema, la presente tesis trata de obtener una solución que es solamente un mínimo local para el problema, y no trata de determinar el mínimo global. *Nótese que el problema de optimización requiere, además de determinar la cantidad mínima de imágenes, determinar la posición que ellas ocupan dentro del espacio paramétrico (esto es, determinar las imágenes de soporte ó puntos de control del modelo) y bajo qué esquema de interpolación entre imágenes se apoya tal solución.*

Dada su naturaleza, este problema no puede ser resuelto por medio de técnicas convencionales de optimización. Una forma de solucionar este problema y que parece ser simple e inmediata, se refiere a construir subconjuntos de combinaciones de imágenes, de entre un conjunto de imágenes del objeto adquiridas previamente, y analizar cada subconjunto de ellas hasta determinar con cuál se obtiene el mínimo global que soluciona el problema. Sin embargo, este proceso combinatorio es evidentemente de una complejidad exponencial en tiempo, y por tanto, ineficiente; además prevalece el problema de determinar qué esquema de interpolación apoyará tal solución. Más aún, se desconoce la cantidad de imágenes que integrarán el conjunto inicial de imágenes sobre el cual procederá el proceso combinatorio. En los siguientes Capítulos se analizará este problema en detalle y se planteará una posible solución para obtener un mínimo local. Se observará que con esta solución, típicamente se obtiene una importante reducción en la cantidad de imágenes requeridas para modelar la apariencia de un objeto cualquiera. Determinar un mínimo global es aún tema de investigación activa.

# CAPÍTULO 2

## *PRELIMINARES MATEMÁTICOS*

**Resumen:** En este Capítulo se exponen algunos conceptos matemáticos, que son necesarios para dar a la presente tesis un fundamento teórico apropiado, y para comprender correctamente el material contenido en los siguientes Capítulos.

---

### 2.1 Definiciones

La denominada luz visible (ó simplemente luz), es una forma de energía que determina un intervalo específico dentro del espectro electromagnético de energía. Esta energía es despedida hacia el espacio por objetos, denominados emisores de luz, los cuales son capaces de generarla. Como ejemplos de este tipo de objetos se puede citar a nuestro sol, y en general otras muchas estrellas en el universo, lámparas (sordas, de aceite, etc.), un objeto en combustión, etc. Esta luz al viajar por el espacio puede incidir sobre otros objetos que no son emisores y ser reflejada en alguna proporción por ellos, nuevamente hacia el espacio. Algunos objetos, denominados sensores, como el ojo humano, las cámaras fotográficas, las cámaras de vídeo, las cámaras digitales, etc., son capaces de registrar la intensidad de la luz en una porción del espacio. El tamaño de esta porción del espacio está determinado por las características propias del sensor empleado.

Usualmente el sensor registra la intensidad de la luz que incide sobre una parte de un plano (usualmente un rectángulo) situado en el espacio. Este plano está determinado por la posición del sensor en el espacio. Un punto específico dentro de este rectángulo, es situado mediante un sistema de coordenadas cartesiano bidimensional, y por tanto su posición es representada como  $(x, y)$ . Entonces, la intensidad de la luz en cualquier punto  $(x, y)$  dentro del rectángulo será una función  $f(x, y)$  que depende de la cantidad de luz incidente sobre los objetos de interés (iluminación) y la cantidad de luz reflejada por dichos objetos (reflectancia). A esta función  $f(x, y)$  definida dentro del rectángulo mencionado, se le conoce como una imagen. Este concepto de imagen es establecido formalmente, mediante la siguiente definición.

**Definición 1.-** El término imagen se refiere a una función bidimensional  $f(x, y)$  de intensidad de luz, donde el valor o amplitud de  $f$  en las coordenadas espaciales  $(x, y)$  da la intensidad (iluminación) de la imagen en ese punto [Gon96]. Usualmente

$x, y \in \mathbf{R}^+ \cup 0$ . Debido a que la luz es una forma de energía,  $f(x, y)$  debe ser no-cero y finita, es decir:  $0 < f(x, y) < \infty$ .

**Definición 2.-** Una imagen digital  $f(x, y)$  es una imagen que ha sido muestreada espacialmente y cuantificada en niveles de gris (ó color), de tal manera que  $x, y, f(x, y) \in \mathbf{Z}^+ \cup 0$  [Go96]. Usualmente,  $0 \leq x \leq M, 0 \leq y \leq N$  y  $L_{\min} \leq f(x, y) \leq L_{\max}$ , donde  $M$  y  $N$  determinan el tamaño de la imagen digital. Cada punto  $(x, y)$  es denominado un pixel,  $f(x, y)$  es la intensidad del pixel  $(x, y)$  y  $L_{\min}$  y  $L_{\max}$  son los niveles mínimo y máximo, respectivamente, de la intensidad (ó tonos de gris ó color) de cualquier pixel de la imagen.

En lo que resta del documento, cualquier referencia al término imagen, se sobreentenderá que se trata de una imagen digital (en tonos de gris), por simplicidad.

**Definición 3.-** Una trayectoria en  $\mathbf{R}^n$  es una función  $\sigma: [a, b] \rightarrow \mathbf{R}^n$  [Ma76].

**Definición 4.-** Sea  $\sigma_1: [a, b] \rightarrow \mathbf{R}^n$  una trayectoria. Sea  $C = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$  tal que  $\sigma_1(x_i) = y_i$ , para  $i = 1, \dots, m$ , con  $x_1 \leq x_2 \leq \dots \leq x_m$ . Se dice que una trayectoria  $\sigma_2: [a, b] \rightarrow \mathbf{R}^n$ , interpola a  $\sigma_1(x)$  dentro del intervalo  $[x_1, x_m]$ , si  $\sigma_2(x_i) = y_i$ , para  $i = 1, \dots, m$ . Usualmente,  $\sigma_2(x)$  es usada para estimar los valores que  $\sigma_1(x)$  toma en el intervalo  $[x_1, x_m]$ . Sin embargo,  $\sigma_2(x)$  no está determinada únicamente. Por esta razón, para un  $\epsilon$  dado,  $\sigma_2(x)$  es seleccionada tal que  $|\sigma_1(x) - \sigma_2(x)| < \epsilon$ , para todo  $x$  en  $[x_1, x_m]$ . Tal  $\epsilon$  es llamado el error asociado a la interpolación.

Si la trayectoria  $\sigma_2(x)$  que interpola a  $\sigma_1(x)$  es una línea recta, la interpolación es lineal; de otra forma, es no-lineal. Cuando la dimensión es  $n=1$ , es común usar polinomios (De Taylor, Lagrange, Hermite, etc.) en una variable [Bu85], para realizar la interpolación.

Algunas veces, es imposible encontrar una trayectoria  $\sigma_2(x)$  tal que cumpla el criterio de error dado dentro del intervalo. En este caso, el problema de interpolación de  $\sigma_1(x)$  es transformado al problema de interpolación segmentaria (ó a pedazos) de  $\sigma_1(x)$ : es necesario encontrar un conjunto de  $m-1$  trayectorias  $\sigma_2(x), \sigma_3(x), \dots, \sigma_m(x)$ , tales que interpolan a  $\sigma_1(x)$ , respectivamente, en los intervalos  $[x_1, x_2], [x_2, x_3], \dots, [x_{m-1}, x_m]$ , dentro de la tolerancia  $\epsilon$ .

La interpolación es empleada cuando la trayectoria  $\sigma_1(x)$  es difícil de evaluar; y por tanto, se busca una trayectoria  $\sigma_2(x)$  que pueda ser evaluada más fácilmente. La interpolación es usada también cuando no se tiene una representación explícita de  $\sigma_1(x)$ , pero se tiene un grupo de valores que adopta la trayectoria dentro de un intervalo.

**Definición 5.-** La Suma de Diferencias Cuadradas (SSD, de sus siglas en inglés) entre dos vectores  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  y  $\mathbf{y} = (y_1, y_2, \dots, y_n)$  en  $\mathbf{R}^n$ , se define como:

$$\|\mathbf{x} - \mathbf{y}\|^2 = \sum_1^n (x_i - y_i)^2$$

Una trayectoria puede ser definida por más de un parámetro. Debido a esto, es necesario generalizar el concepto de trayectoria dado en la Definición 3. Por supuesto, en este caso, la imagen de la trayectoria no será una curva, sino una superficie, por lo que entonces definimos una superficie parametrizada como:

**Definición 6.-** Una superficie parametrizada es una función  $\phi: D \rightarrow \mathbf{R}^M$ , donde D es algún dominio en  $\mathbf{R}^N$  [Ma76].

Así mismo, es necesario generalizar el concepto de interpolación para una superficie parametrizada.

**Definición 7.-** Sea  $\phi_1: D \subset \mathbf{R}^N \rightarrow \mathbf{R}^M$  una superficie parametrizada. Sea  $C = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_k, \mathbf{y}_k)\}$  tal que  $\phi_1(\mathbf{x}_i) = \mathbf{y}_i$ , para  $i=1, \dots, k$ . Se dice que una superficie parametrizada  $\phi_2: D \subset \mathbf{R}^N \rightarrow \mathbf{R}^M$ , interpola a  $\phi_1(\mathbf{x})$  en los puntos  $\mathbf{x}_i$  en C,  $i=1, \dots, k$ , si  $\phi_2(\mathbf{x}_i) = \mathbf{y}_i$ , para  $i=1, \dots, k$ . Debido a que  $\phi_2(\mathbf{x})$  no está determinada únicamente, usualmente, para un  $\epsilon$  dado,  $\phi_2(\mathbf{x})$  es seleccionada tal que  $|\phi_1(\mathbf{x}) - \phi_2(\mathbf{x})| < \epsilon$ , para todo  $\mathbf{x}$  en D. Tal  $\epsilon$  es llamado el error asociado a la interpolación.

De manera similar al caso de trayectorias, algunas veces es imposible encontrar una superficie  $\phi_2(\mathbf{x})$  tal que cumpla el criterio de error dado dentro de un dominio. En este caso, el problema de interpolación de  $\phi_1(\mathbf{x})$  es transformado al problema de interpolación segmentaria (ó a pedazos) de  $\phi_1(\mathbf{x})$ : es necesario encontrar un conjunto de l-1 trayectorias  $\phi_2(\mathbf{x}), \phi_3(\mathbf{x}), \dots, \phi_l(\mathbf{x})$ , tales que interpolan a  $\phi_1(\mathbf{x})$ , respectivamente, en una partición del dominio, dentro de la tolerancia  $\epsilon$ .

## 2.2 Adquisición uniforme de imágenes

Para crear modelos basados en apariencia, se comienza con la adquisición de imágenes del objeto a modelar. Este proceso es independiente de la técnica que se emplee posteriormente para reconocer al objeto. La adquisición es realizada tomando imágenes alrededor del objeto, sobre un plano (el plano de la tornamesa). El objeto rota frente a la cámara (o la cámara rota alrededor del objeto) por un ángulo  $\theta_i$ , respecto a una posición inicial fija para adquirir la i-ésima imagen  $\mathbf{I}_i$ , con  $i=1, \dots, r$  y  $\theta_{i-1} < \theta_i$ . Esta forma de adquirir las imágenes define un parámetro (el ángulo de giro) para la tarea de reconocimiento del objeto. Hay varios enfoques para construir posteriormente los modelos. Uno de ellos usa las imágenes completas para formar los modelos. Otros usan solamente algunas características extraídas de las imágenes, las

cuales se piensa describen al objeto. En este trabajo doctoral, el modelo es construido empleando las imágenes completas.

Para realizar la adquisición de imágenes, es necesario un sistema para rotar el objeto sobre un plano (plano de la tornamesa). Un esquema de tal sistema es mostrado en la Figura 3 (Capítulo 1). Aunque la fuente de iluminación y la cámara pueden adoptar cualquier posición respecto al objeto, aquí se supondrá que ocupan una posición fija, manteniendo estas suposiciones durante los siguientes Capítulos 3 y 4, para posteriormente remover estas suposiciones en el Capítulo 5. Esto indica que en los 2 capítulos señalados se supondrá que la adquisición de imágenes estará definida por un solo parámetro, determinado por el ángulo de giro del objeto frente a la cámara o sensor, y posteriormente en el Capítulo 5 se abordará el problema de la adquisición de imágenes, cuando tal proceso depende de N-parámetros.



Figura 6: Objeto empleado para adquirir la secuencia de imágenes mostrada en la Figura 7.

Si las imágenes  $I_1, \dots, I_r$  adquiridas alrededor del objeto son representadas como matrices de  $n \times m$ , y las columnas de cada matriz son empiladas para formar vectores  $v_1, \dots, v_r \in \mathbf{R}^{nm}$ , entonces, observando la Definición 3, se puede notar que tales vectores pertenecen a una trayectoria:  $\sigma: [0, 2\pi] \rightarrow \mathbf{R}^{nm}$  parametrizada por el ángulo de rotación  $\theta$  del objeto frente a la cámara, respecto a la posición inicial fija. Esta trayectoria es conocida en la literatura como **la trayectoria determinada por la apariencia del objeto**, o simplemente, **la apariencia del objeto**.

Usualmente, las técnicas propuestas para el reconocimiento de objetos, suponen, por simplicidad, que  $\theta_i - \theta_{i-1} = k$ , con  $k$  una constantes, esto es, el objeto rota (sobre el plano de la tornamesa) siempre el mismo ángulo (con respecto a la posición actual), para capturar la siguiente imagen [St00]. Un ejemplo de esta forma de adquirir imágenes es mostrado en la Figura 7, para el objeto ilustrado en la Figura 6, usando un espaciamiento en el ángulo de giro de la tornamesa entre imágenes consecutivas de  $10^\circ$ . Valores típicos encontrados en la literatura para  $k$  (ángulo de giro de la tornamesa entre imágenes consecutivas), oscilan entre  $5^\circ$  y  $12^\circ$ . Estos valores son basados en observaciones empíricas, y por lo tanto no presentan formalidad alguna para suponerlos. Más aún, este valor es fijo e independiente de la clase de objeto (textura, geometría, color, etc.). Esto implica que la cantidad de imágenes es fija para cualquier



objeto. Esta suposición es errónea (como se verá posteriormente en el Capítulo 4), debido a que el número de imágenes debe ser dependiente de la apariencia del objeto. Se puede comprobar este hecho en ejemplos extremos (vasos, copas, frutas, etc.): uno de estos casos es una esfera coloreada homogéneamente, por supuesto, el ejemplo más extremo. En este caso, se requiere una única imagen para modelarla, debido a que un solo punto en un espacio multidimensional constituye al modelo.

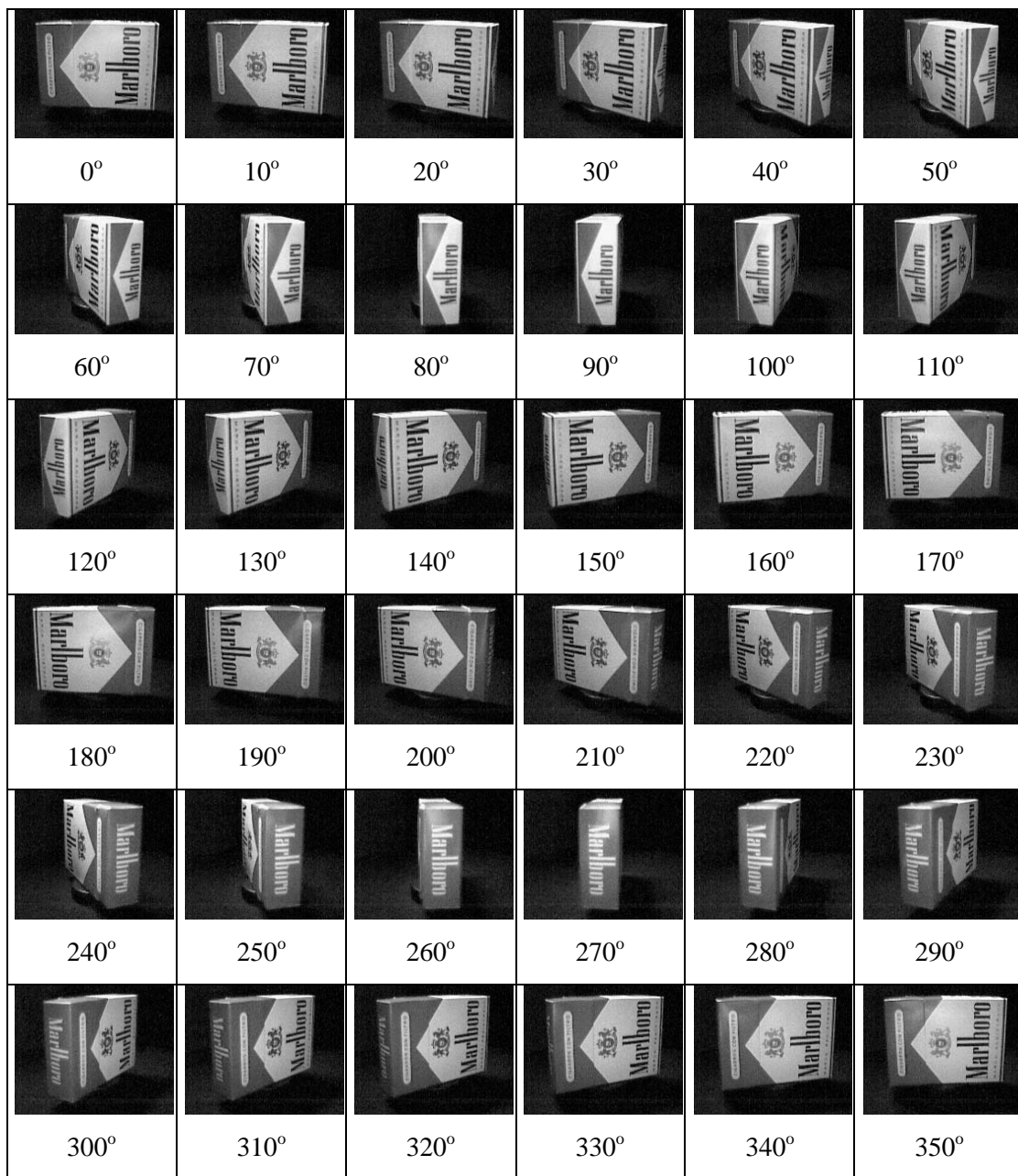


Figura 7.- Adquisición de imágenes para el objeto mostrado en la Figura 6. El ángulo de giro de la tornamesa entre imágenes consecutivas es de 10°.

Adquirir imágenes en una forma uniforme supone una capacidad de almacenamiento grande para el sistema de modelado. Así mismo, en técnicas que requieren un cálculo intenso como la técnica de espacios propios [Mu92], considerar una cantidad mayor de imágenes que las estrictamente necesarias, implica un incremento considerable en el tiempo requerido para el cálculo del modelo.

### 2.3 Reconocimiento de objetos usando espacios propios (*eigenspaces*)

Como se ha explicado brevemente en la introducción, la técnica de espacios propios permite reconocer cualquier tipo de objetos, si se le proporcionan las imágenes necesarias para capturar la apariencia del objeto. Enseguida se expone el proceso de modelado y reconocimiento de objetos.

Como se mencionó en la sección previa, y se puede observar en [Mu92] y [Mu95], la técnica de espacios propios de Murase y Nayar parte de la adquisición de un conjunto de imágenes del objeto de interés, alrededor de él, y considerando probablemente diferentes condiciones de iluminación, como se explicó en la Sección previa. Sea  $\{\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_M\}$  el conjunto de imágenes adquiridas. Este conjunto puede provenir de un muestreo uniforme o no-uniforme del espacio visual de trabajo. En [Mu95] fue usado un muestreo uniforme.

Las imágenes en el conjunto anterior están correlacionadas en un alto grado si el desplazamiento entre imágenes consecutivas es pequeño. La técnica de espacios propios explota este hecho, comprimiendo el conjunto de imágenes en una representación con pocas dimensiones, debido a que el número de dimensiones depende inversamente de la correlación entre las imágenes. Además, esta compresión captura las características principales de la apariencia del objeto. Esto es realizado por medio del uso del Análisis de Componentes Principales [Oja83], donde los vectores propios del conjunto de imágenes son calculados y usados como una base ortogonal para representar las imágenes individuales.

Para lograr invariancia en escala, todas las imágenes adquiridas son forzadas a tener el mismo tamaño (escalando si es necesario). También cada imagen es normalizada en brillantez e intensidad usando:  $\mathbf{i}'_j = \mathbf{i}_j / \|\mathbf{i}_j\|$ . Sea  $\{\mathbf{i}'_1, \mathbf{i}'_2, \dots, \mathbf{i}'_M\}$  el conjunto de imágenes obtenido después de esta operación.

Enseguida, la matriz de promedios  $\mathbf{c}$  de todas las imágenes en el conjunto (que es la matriz cuyo elemento  $c_{ij}$  es obtenido al promediar aritméticamente los valores de los correspondientes píxeles  $ij$ -ésimos, de todas las imágenes), es restada a cada imagen. Esto asegura que el vector propio con el mayor valor propio representará la dimensión del subespacio en el cual la varianza de las imágenes será mayor en el

sentido de correlación. Una matriz de imágenes es construida sustrayendo  $c$  de cada imagen y empilando las imágenes por columnas para obtener:

$$\mathbf{P} = \{ \mathbf{i}'_1 - \mathbf{c}, \mathbf{i}'_2 - \mathbf{c}, \dots, \mathbf{i}'_M - \mathbf{c} \}$$

$\mathbf{P}$  es una matriz de  $N \times M$ , donde  $N$  es el número de píxeles en cada imagen y  $M$  es el número total de imágenes en el conjunto. Para calcular los vectores propios del conjunto de imágenes, se define la matriz de covarianza:  $\mathbf{Q} = \mathbf{P} \mathbf{P}^T$ .  $\mathbf{Q}$  es de  $N \times M$  y es una matriz muy grande debido a que un gran número de píxeles constituyen a cada imagen.

Los vectores propios  $\mathbf{e}_k$  y sus correspondientes valores propios  $\lambda_k$  de  $\mathbf{Q}$  son determinados a través de resolver el problema bien conocido:

$$\lambda_k \mathbf{e}_k = \mathbf{Q} \mathbf{e}_k$$

El cálculo de los vectores propios de una matriz tan grande como  $\mathbf{Q}$  es computacionalmente intensa [Na96a] y depende del número de imágenes de entrenamiento si esta cantidad es menor que el número de píxeles que constituyen a las imágenes; de otra forma, depende del número de píxeles.

Al resolver el problema anterior se obtiene un conjunto de valores propios

$$\{ \lambda_k \mid k = 1, 2, \dots, K \} \text{ donde } \{ \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_K \}$$

y su correspondiente conjunto de vectores propios ortonormales

$$\{ \mathbf{e}_k \mid k = 1, 2, \dots, K \}$$

Estos  $K$  vectores propios constituyen el espacio propio (con  $K \ll N$ ), el cual es una aproximación al espacio propio completo con  $N$  dimensiones.

Cada imagen en el conjunto de imágenes de muestra  $\mathbf{i}'_j$  es proyectada en el espacio propio con:

$$\mathbf{f}_j = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_K]^T (\mathbf{i}'_j - \mathbf{c})$$

para obtener el punto  $\mathbf{f}_j$ . Estos puntos discretos son interpolados para obtener una trayectoria (*manifold*), que representa la apariencia del objeto para todas las posibles posiciones de la cámara alrededor del objeto. En [Mu95] tal interpolación fue hecha a través de Splines. Esta trayectoria reside en un espacio de pocas dimensiones, y es una representación continua y compacta de la apariencia del objeto.

Finalmente, para la etapa de reconocimiento, se obtiene una nueva imagen  $\mathbf{i}'_c$  tomada en cualquier posición de la cámara (dentro del rango permitido), y se proyecta esta imagen en el espacio propio, para obtener el punto:

$$\mathbf{f}_c = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_K]^T (\mathbf{i}'_c - \mathbf{c})$$

y entonces el reconocimiento propiamente dicho se reduce a localizar la trayectoria  $q$  que esté más cercana al punto  $\mathbf{f}_c$ :

$$d = \min_q \| \mathbf{f}_c - \mathbf{f}(q) \|$$

y entonces se dice que la imagen  $\mathbf{i}'_c$  pertenece al objeto  $q$ .

# CAPÍTULO 3

## ***MODELADO Y RECONOCIMIENTO MEJORADO DE OBJETOS***

**Resumen:** Como se mencionó en la introducción, la presente tesis plantea mejorar la precisión de los modelos y reducir la cantidad de imágenes necesaria para reconocer objetos. En el presente capítulo se hace un planteamiento formal del problema a resolver.

---

### **3.1 Requerimientos de almacenamiento**

En esta sección se realizará el cálculo del número de imágenes requeridas para capturar la apariencia de un objeto, cuando se tienen  $N$  parámetros permitidos como posibles variables para la apariencia del objeto. Aquí se asume que el muestreo es uniforme para cada parámetro.

Como se mencionó en la sección previa, el cálculo depende del valor de incremento del ángulo que se maneje para cada parámetro. Sin pérdida de generalidad, supóngase que para capturar la apariencia del objeto para el primer parámetro (que aquí se supondrá será el ángulo de giro de la tornamesa frente al sensor), el muestreo propuesto (que típicamente oscila entre 5-12 grados y divide exactamente a 360), es  $k_1$ . Entonces, el número total de valores discretos para el primer parámetro es:

$$I_1 = 360 / k_1 \quad (1)$$

que además será el número de imágenes

$$T_1 = I_1 \quad (2)$$

que se requieren para cubrir el primer parámetro. El subíndice indica el parámetro que se está cubriendo. De manera similar, si para capturar la apariencia del objeto en cuestión se definiera un segundo parámetro, y el muestreo propuesto para ese parámetro fuera  $I_2 = 360 / k_2$ , bastaría multiplicar la expresión (2) por  $I_2$  para obtener el número total de imágenes requeridas,  $T_2 = I_1 I_2$ . Siguiendo este procedimiento, se obtiene la fórmula general para la cantidad de imágenes requeridas, para capturar la apariencia de un objeto, bajo  $N$  parámetros:

$$T_N = I_1 I_2 \dots I_N \quad (3)$$

Supóngase, para propósitos ilustrativos, que  $k_j = 10$ , para  $j=1, \dots, N$ . Entonces la expresión (3) se reduce a:  $T_N = 36^N$  (4)

En la Tabla 1 se muestra el número de imágenes requerido, según la expresión (4), para algunos valores de N.

	Parámetros (N)				
	1	2	3	4	5
Número de imágenes (T <sub>N</sub> )	36	1296	46656	1679616	60466176

Tabla 1.- Tabulación del número de imágenes requeridas para capturar la apariencia de un objeto, en función del número de parámetros.

De acuerdo a los valores mostrados en la Tabla 1, se puede observar que la cantidad de imágenes requeridas para capturar la apariencia de un objeto, crece enormemente si se incrementa el número de parámetros definidos para capturar su apariencia. Nótese que se han tabulado valores hasta para 5 parámetros. Sin embargo, esta cantidad de parámetros no es difícil de alcanzar e inclusive de superar: basta considerar objetos articulados, en donde sus articulaciones definen más de 5 parámetros. Ejemplos de este tipo de objetos son: el cuerpo humano, el cuerpo de algunos animales, algunos juguetes, etc.

Para obtener finalmente los requerimientos de almacenamiento del sistema, basta multiplicar la cantidad de imágenes requeridas, por el número de bytes que necesita una imagen para ser almacenada por el sistema. Esto por supuesto, depende también del formato en el que el sistema almacena las imágenes, y aquí no se ahondará más en esto. De cualquier manera, la cantidad de bytes requeridos es asombrosamente grande, y es requerida forzosamente para modelar al objeto, pues según lo discutido en la Sección 2.3, la técnica de espacios propios, en particular, requiere la presencia simultánea de todas las imágenes que capturan la apariencia del objeto, para poder ser aplicado. Es por ello, que los requerimientos de almacenamiento del sistema deberán ser extremadamente grandes.

### 3.2 Requerimientos de tiempo de procesamiento

Evidentemente, si la cantidad de imágenes que deberá procesar el sistema que construye el modelo del objeto se incrementa, se deberá incrementar también el tiempo de cómputo requerido para construir tal modelo. Esto se concluye porque según la técnica de espacios propios (Ver [Mu92]), las imágenes que se le proporcionan al sistema de reconocimiento son transformadas a una representación de vectores columna, y con ellos se construye la matriz de covarianza, para la cual se calcularán los vectores y valores propios que serán empleados para construir el modelo. Como es bien sabido, la complejidad de los algoritmos empleados para calcular tales valores y

vectores propios depende del tamaño de la matriz, por lo que de aquí se obtiene que entre más grande sea el tamaño de la matriz, mayor será el tiempo de cómputo.

Para dar una idea del tiempo de cómputo requerido, se citan los resultados obtenidos por Nayar, Murase y Nene en [Na96a], para el cálculo del modelo de un objeto rígido: en una estación de trabajo Sun IPX, 20 vectores propios de un conjunto de 100 imágenes (cada una de 128x128 píxeles) pueden ser calculados en alrededor de **3 minutos**, y 20 vectores propios de un conjunto de 1000 imágenes (cada una de 128x128 píxeles) en aproximadamente **4 horas**. Obsérvese el incremento considerable en el tiempo de cómputo requerido. Aunque no se ahondará en esto, no es difícil observar de estos resultados, que para calcular los vectores propios para la cantidad de imágenes listadas en la Tabla 1, el tiempo requerido será extremadamente grande, y por tanto, prohibitivo para la mayoría de las computadoras actuales.

### **3.3 Reducción de la cantidad de imágenes**

Por lo planteado en las secciones previas, se puede observar que, si se desea que un sistema de reconocimiento que maneja modelos basados en apariencia sea eficiente, una de las estrategias que se desearía desarrollar sería un procedimiento que reduzca la cantidad de imágenes que se requieren para construir tales modelos, y que fueron mostradas en la Tabla 1. Tal reducción implicará inmediatamente que los requerimientos tanto de tiempo como en espacio se reducirán también, y esto generará sistemas de reconocimiento más eficientes. Esta aseveración en particular será cierta para la técnica de espacios propios de Murase y Nayar.

El objetivo central del presente trabajo de tesis es entonces reducir la cantidad de imágenes necesarias para construir modelos basados en apariencia. Esta reducción será además acompañada de la garantía de que el conjunto reducido de imágenes, permitirá la construcción de los modelos con una precisión  $\epsilon$  preestablecida.

Esta reducción se logrará mediante el empleo de una técnica sistemática de muestreo no uniforme que llevará un control de la precisión del modelo y que determinará cuáles y cuántas imágenes son estrictamente necesarias para construir el modelo, para una precisión  $\epsilon$  preestablecida. Esta técnica será expuesta en detalle en el Capítulo 4.

La reducción de la cantidad de imágenes es lograda por medio de explotar la siguiente idea central: los sistemas propuestos en la literatura (Ver [Al99] para una exposición detallada de tales técnicas) suponen, por simplicidad, que una cantidad fija de imágenes es requerida para capturar la apariencia de cualquier objeto. Esto significa que tales sistemas adquieren las imágenes con un muestreo uniforme, adoptando un ángulo de giro constante entre imágenes sucesivas. Esto a su vez muestra que no existe ningún control acerca de la precisión con que se construyen los modelos. Sin

embargo, tal suposición es totalmente errónea y no presenta ninguna formalidad ni precisión en los resultados obtenidos. Esto se sigue del hecho de observar que existe una gran cantidad de objetos que requieren menos cantidad de imágenes para construir sus modelos. Quizá el ejemplo más extremo sea una esfera pintada homogéneamente de un determinado color. Su modelo será un solo punto en un espacio de varias dimensiones, pues todas las imágenes que se le tomen alrededor serán iguales. Otra clase de objetos que probablemente requieran menor cantidad de imágenes serán aquellos que posean simetría central (respecto a la cámara) y que posean una textura casi homogénea en algunas de sus partes; ejemplos de este tipo de objetos serán: vasos, frascos, etc. De igual manera, existen clases de objetos que debido a su complejidad geométrica y/o complejidad de su textura, requieran una mayor cantidad de imágenes para capturar su apariencia, para una determinada precisión  $\epsilon$  preestablecida. Estas propiedades son bien conocidas y citadas en la literatura [Sh99], pero los sistemas actuales no las explotan.

Entonces, como se puede observar, dado que la suposición de un muestreo uniforme es errónea, el desarrollo de una técnica adaptativa que dependa del tipo de objeto a modelar para determinar cuántas y cuáles imágenes son estrictamente necesarias para construir el modelo, repercutirá en dos aspectos fundamentales:

- i) mejora en la precisión de los modelos, y/o
- ii) reducción tanto de tiempo de cómputo, como de recursos de almacenamiento.

Entonces, *el objetivo de la presente tesis es el de desarrollar tal técnica adaptativa que cubra los dos aspectos señalados.*

A la fecha, ningún trabajo previo ha hecho una propuesta para guiar el proceso de adquisición de imágenes. Todos los trabajos previos emplean el método de ensayo-error, para adquirir las imágenes que se requieren para reconocer a los objetos (tanto rígidos como articulados): asumen un espaciamiento fijo para muestrear el espacio de trabajo (normalmente esto lo hacen suponiendo un incremento fijo, en el ángulo de giro de la tornamesa, y en el ángulo de elevación del sensor), y realizan una serie de ensayos experimentales para determinar la calidad de los resultados obtenidos. Si éstos son pobres, reducen el incremento en el espaciamiento, y realizan una serie más de experimentos, y continúan este proceso hasta obtener resultados aceptables. Se refiere al lector a revisar, por ejemplo, los trabajos de Murase y Nayar [Na96a], [Mu92], los trabajos de Nelson [Ne97], [Ne98a], y los trabajos de Pauli [Pa95].

# CAPÍTULO 4

## *ADQUISICIÓN NO-UNIFORME DE IMÁGENES*

**Resumen:** Como se mencionó en el capítulo anterior, el objetivo de la presente tesis se centra en la reducción de la cantidad de imágenes requerida para construir modelos basados en apariencia. Esta reducción deberá ser acompañada en una mejora en la precisión de los modelos obtenidos. Se propone que tal reducción se realice mediante el desarrollo de una técnica adaptativa que realice automáticamente el proceso de adquisición de imágenes para cada objeto en particular. En este capítulo se plantea el desarrollo de tal técnica, su justificación, experimentos realizados con tal técnica y que justifican su aplicación junto con una discusión sobre aspectos importantes derivados del empleo de tal técnica.

---

### **4.1 Adquisición adaptativa (guiada por interpolación lineal y SSD)**

Como se señaló en la Sección 3.1, adquirir imágenes de una manera uniforme supone una capacidad de almacenamiento grande para el sistema de modelado. Así mismo, en técnicas que requieren un cálculo intenso como la técnica de espacios propios, considerar una cantidad mayor de imágenes que las estrictamente necesarias, implica un incremento considerable en el tiempo requerido para el cálculo del modelo.

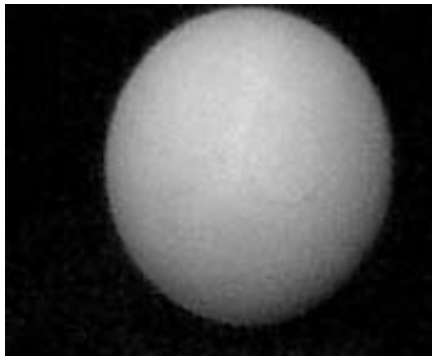


Figura 8: Un objeto para el cual la reducción en la cantidad de imágenes es extrema: una pelota.



Más aún, como se señaló en la Sección 3.3, es un error suponer que la cantidad de imágenes requerida para modelar un objeto es independiente de la clase de objeto a reconocer. Como un ejemplo extremo, quizá el más extremo, obsérvese el objeto mostrado en la Figura 8 (pelota). Desde cualquier ángulo que se le observe (empleando una iluminación difusa), la imagen adquirida por la cámara será la misma, salvo por pequeñas imperfecciones que pudiera haber en su superficie, que no alterarán en gran medida la similitud entre las imágenes. Si tales imperfecciones no existiesen, esto es, la superficie del objeto estuviera perfectamente pulida, entonces todas las imágenes serían iguales, y por tanto, el modelo asociado al objeto deberá ser un único punto en un espacio multidimensional. Esta propiedad es bien conocida en la literatura [Sh99], pero los sistemas de reconocimiento que han sido propuestos, no la explotan y por tanto, la cantidad de imágenes se vuelve prohibitiva (Ver Tabla 1).



Figura 9: Objetos que presentan simetría central respecto al centro de la tornamesa.

Existe otro tipo de objetos que aunque su modelo no será un punto en el espacio multidimensional, si requieren pocas imágenes para poder ser modelados; es el caso de objetos con simetría central y posiblemente, con detalles no significativos en su superficie: ejemplos de este tipo de objetos son vasos, latas, recipientes de muy diversas clases, etc. En la Figura 9 se muestran algunos ejemplos de este tipo de objetos. Obsérvese que los detalles grabados en su superficie son diversos, al igual que los materiales con que están fabricados (vidrio y plástico).

Para estas clases de objetos, es posible lograr una reducción significativa en la cantidad de imágenes que se requiere para poder modelarlos. Más aún, como se mostrará en las secciones siguientes, en general para cualquier tipo de objeto será posible obtener una reducción en la cantidad de imágenes requerida para modelarlos. El otro aspecto significativo consiste en que es posible mejorar la precisión de los modelos; más aún, es posible lograr una precisión deseada para obtener un mejor reconocimiento de los objetos; esto se logrará mediante una técnica que analiza el tipo de objeto que se desea modelar, y en base a las características que presente su apariencia, determinará cuántas y cuáles imágenes son estrictamente necesarias para modelarlo, para una precisión  $\epsilon$  especificada.

La idea central de tal técnica se basa en la observación de que existen zonas en el modelo del objeto que presentan un comportamiento cuasi-lineal, y por tanto,

pueden ser aproximadas por una pequeña cantidad de imágenes, y en las zonas donde el comportamiento no es cuasi-lineal, se requerirá una cantidad de imágenes mayor. Esto significa que la técnica aquí propuesta genera como salida un muestreo no-uniforme que cumple con el requerimiento de precisión  $\epsilon$  proporcionado por el investigador. Esto implica que es posible generar todo un conjunto de modelos para un mismo objeto, que se diferencian uno de otro en la precisión que haya especificado el investigador para cada caso.

Como se señaló en la Sección 2.2, adquirir imágenes en una forma uniforme supone una gran capacidad de almacenamiento para el sistema de modelado. Así mismo, en técnicas que requieren un cálculo intenso como la técnica de espacios propios, se genera un incremento considerable en el tiempo requerido para el cálculo del modelo.

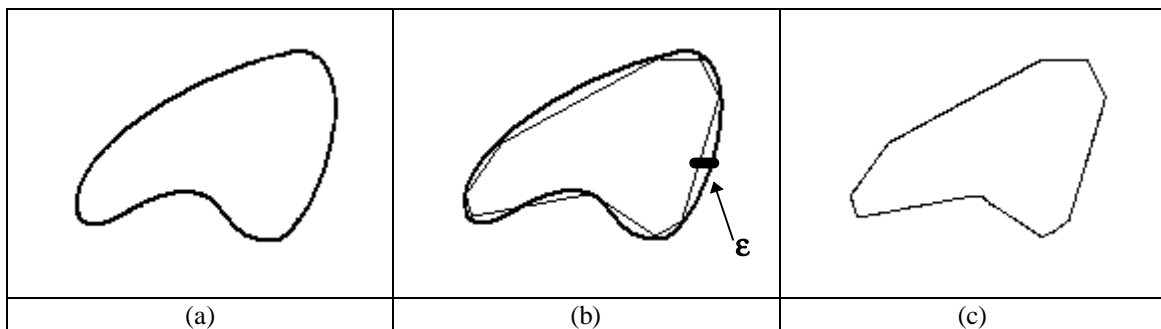


Figura 10: La Figura (a) muestra el comportamiento de la trayectoria determinada por la apariencia del objeto. Aunque la trayectoria realmente reside en un espacio multidimensional, aquí se muestra en un plano para propósitos ilustrativos. En (b) se muestra la interpolación lineal segmentaria para la trayectoria de (a). La diferencia entre la apariencia del objeto (curva oscura) y la estimación por interpolación lineal (curva en líneas claras) es menor que  $\epsilon$  en los puntos medios de cada línea recta. En (c) se muestra el comportamiento estimado por medio de la técnica propuesta. Los puntos de las esquinas señalan a las imágenes estrictamente necesarias para aproximar la apariencia del objeto, dentro de un error  $\epsilon$  dado.

Para superar las anteriores limitaciones, se propone el empleo de un muestreo no-uniforme de imágenes del objeto, soportado por una técnica simple que adapta el valor de los ángulos  $\theta_i$  de incremento en el ángulo de giro de la tornamesa, respecto a una posición inicial fija, de acuerdo a los siguientes criterios:

- Apariencia del objeto, y
- Requerimientos de precisión.

Esta técnica, ilustrada en la Figura 10, está basada en la observación de que la apariencia del objeto puede ser aproximada por medio de una interpolación lineal segmentaria (ver Definición 4), dentro de un intervalo de error  $[-\epsilon, \epsilon]$ .

## 4.2 Algoritmo adaptativo

La técnica inicia con la adquisición de una pareja de imágenes denotadas por  $\mathbf{X}$  y  $\mathbf{Y}$ , tomadas a  $0^\circ$  y  $180^\circ$ , respectivamente. Enseguida, la técnica interpola linealmente entre  $\mathbf{X}$  y  $\mathbf{Y}$  usando la función:

$$g(\lambda) = \lambda\mathbf{X} + (1-\lambda)\mathbf{Y} \quad ; \quad 0 \leq \lambda \leq 1. \quad (5)$$

La Figura 11 muestra un ejemplo del tipo de imágenes obtenidas al aplicar la expresión (5) sobre imágenes de objetos reales. Por supuesto, las imágenes interpoladas no tienen interpretación física, pero aquí son empleadas para aproximar el modelo del objeto.

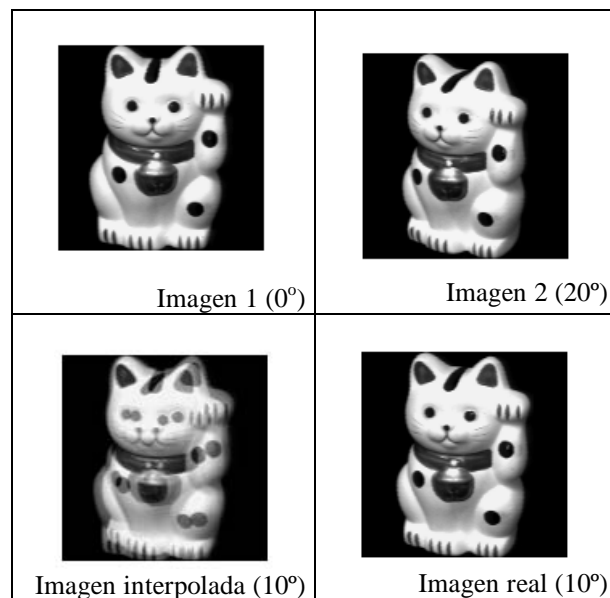


Figura 11: Ejemplo de interpolación de imágenes, mediante el empleo de la expresión (1) aplicada a las imágenes 1 y 2 ( $g(0.5)$ ). La imagen real para el punto correspondiente es también mostrada.

Enseguida, la técnica indica adquirir otra imagen  $\mathbf{Z}$  en el punto medio ( $90^\circ$ ), y comparar  $\mathbf{Z}$  con la imagen  $\mathbf{Z}'$  obtenida por interpolación lineal en el mismo punto, el cual corresponde a  $g(0.5)$ , según la expresión (1). Si la SSD (ver Definición 5) entre  $\mathbf{Z}$  y  $\mathbf{Z}'$  es menor que  $\varepsilon$ , la técnica determina que solo una pareja de imágenes es requerida para aproximar la apariencia del objeto entre  $0^\circ$  y  $180^\circ$ . Si la condición no se cumple, el espacio de trabajo es dividido en dos partes: la primer parte comprende a las imágenes entre  $0^\circ$  y  $90^\circ$ , mientras que la segunda parte comprende a las imágenes entre  $90^\circ$  y  $180^\circ$ . Un análisis similar es realizado para las imágenes comprendidas entre

180° y 360° (note que la imagen a 0° en la misma que la imagen a 360°). Este proceso continúa analizando de la misma forma cada nuevo intervalo generado. El algoritmo termina cuando es posible aproximar la apariencia del objeto dentro de un error  $\epsilon$ . Evidentemente, en la práctica la técnica puede terminar sin poder aproximar la apariencia del objeto, si se rebasa la precisión máxima a que el sistema de hardware puede adquirir imágenes. Esta precisión máxima se define como el mínimo ángulo a que puede girar la tornamesa, entre imágenes consecutivas. Sin embargo, ésta no es una limitante inherente a la técnica, pero debe ser considerada por ella para evitar que se requiera una cantidad infinita de imágenes, y por tanto, nunca finalice.

### Algoritmo 1.

- 0.- Asignar la precisión deseada  $\epsilon$ .
- 1.-  $L_i = 0$ . { límite inferior del intervalo a analizar }
- 2.-  $L_s = 360$ . { límite superior del intervalo a analizar }
- 3.- Adquirir imágenes para las posiciones  $L_i$  y  $L_s$ .
- 4.- Si (  $(L_s - L_i) < 2\theta$  ) hacer **Error = 1 e ir a 15.** { Se rebasó la precisión del hardware }
- 5.-  $L_m = (L_s + L_i)/2$ . { calcular el punto medio del intervalo }
- 6.- Adquirir la imagen correspondiente a la posición  $L_m$ .
- 7.- Interpolar linealmente entre las imágenes  $L_i$  y  $L_s$ , usando la expresión (5).
- 8.- Si es posible interpolar linealmente a la imagen  $L_m$  con las imágenes  $L_i$  y  $L_s$ , dentro de un margen de error  $\epsilon$ , usando  $g(0.5)$  en la expresión (1), y usando SSD como criterio para comparar  $L_m$  y  $g(0.5)$ , esto es,  

$$\|L_m - g(0.5)\|^2 \leq \epsilon$$
entonces ir al paso 12.
- 9.- Almacenar en una pila el intervalo ( $L_m, L_s$ ).  
{ el intervalo ( $L_m, L_s$ ) será analizado posteriormente }
- 10.-  $L_s = L_m$ . { reducción del intervalo a analizar }
- 11.- Ir a 4. { nuevo análisis con un intervalo menor }
- 12.- Guardar las imágenes  $L_i$  y  $L_s$  como imágenes necesarias para construir el modelo del objeto.
- 13.- Si la pila no está vacía, extraer de ella una pareja de valores y asignarla a ( $L_i, L_m$ ), e ir al paso 4.
- 14.- **Error = 0**
- 15.- **Fin.**

El algoritmo 1 formaliza las ideas previamente expuestas. Por simplicidad, el análisis comienza con un solo intervalo comprendido entre 0° y 360°, pero nótese que esas imágenes son iguales, entonces, el algoritmo realmente debería comenzar analizando los intervalos 0° a 180°, y 180° a 360°. La precisión máxima del hardware se asume que es igual a  $\theta$ . Nótese que en el paso 4 se verifica si la longitud del intervalo a analizar es superior a  $2\theta$  en lugar de  $\theta$ ; esto se debe a que el algoritmo requiere que el hardware pueda adquirir al menos una imagen, dentro del intervalo que

analiza (diferente a los extremos) para verificar la precisión del modelo en ese intervalo. Si la precisión máxima del hardware es alcanzada, el algoritmo informa esta condición regresando la variable  $Error=1$ , indicando que la precisión  $\epsilon$  no puede ser alcanzada con la precisión máxima del hardware. Si  $Error=0$ , el algoritmo entrega el modelo estimado para la precisión  $\epsilon$  pedida.

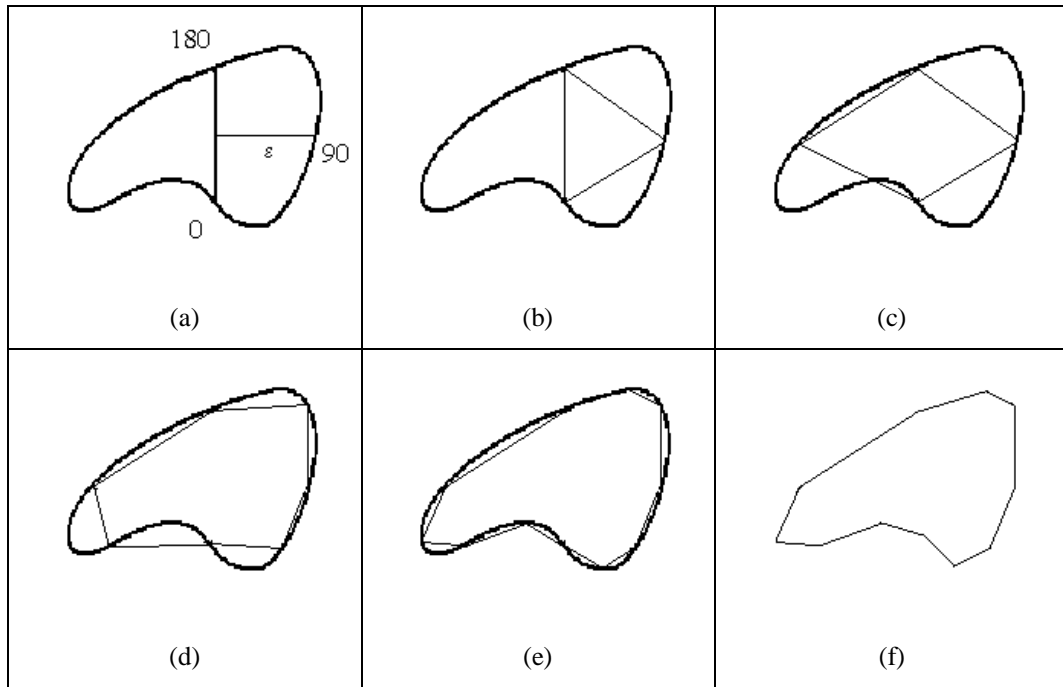


Figura 12: Una vista esquemática de cómo el algoritmo propuesto logra aproximar la apariencia del objeto, mediante interpolación lineal segmentaria (ver texto para explicación).

Un ejemplo esquemático de cómo el algoritmo propuesto logra aproximar la apariencia del objeto, es mostrado en la Figura 12. Aquí se ilustra el comportamiento del algoritmo en dos dimensiones para propósitos ilustrativos, sin embargo, recuérdese que la apariencia de un objeto reside en un espacio de más de tres dimensiones, por lo que no posee interpretación física. En cada una de las Figuras 12a-e se ilustran pasos consecutivos del algoritmo, respectivamente. La apariencia del objeto se ilustra mediante líneas oscuras, y las aproximaciones sucesivas que va construyendo el algoritmo propuesto son ilustradas mediante líneas claras. En la Figura 12a se señala la primera aproximación como dos líneas juntas dado que el análisis comienza en los intervalos,  $0^{\circ}$ - $180^{\circ}$ ,  $180^{\circ}$ - $360^{\circ}$ , y la longitud de la línea etiquetada con  $\epsilon$  (que une un punto de la aproximación con el correspondiente punto en la apariencia del objeto), es el criterio que sigue el algoritmo propuesto, para determinar si se requiere adquirir más imágenes para modelar al objeto en esa zona, bajo la precisión  $\epsilon$ : si la longitud de

esta línea es mayor a  $\epsilon$ , se adquiere una imagen más (Figura 12b); de otra manera, el algoritmo termina de analizar esa zona. Finalmente, en la Figura 12f se muestra la aproximación lograda por el algoritmo propuesto.

### 4.3 Muestreo no-uniforme y espacios propios

Previamente se mencionó que la técnica de espacios propios (EP) es computacionalmente intensa, y por lo tanto, nuestra técnica propuesta puede ser usada para reducir el tiempo requerido para calcular el modelo del objeto y para mejorar la precisión de tal modelo.

Pero más aún, EP usa *splines* para aproximar la apariencia del objeto. Un *spline* es una forma de interpolación no-lineal [Bu85], computacionalmente más cara que la interpolación lineal, y por lo tanto, el uso de *splines* debe realizarse teniendo certidumbre de que las imágenes adquiridas son estrictamente indispensables para construir el modelo del objeto. Tal certidumbre es proporcionada por el algoritmo propuesto, teniendo la ventaja adicional de un costo computacional menor, dado que se basa en interpolación lineal. Por supuesto, en general, cualquier forma de interpolación no-lineal puede ser empleada por nuestra técnica, pero el costo será mayor.

Con base en lo anterior y a diferencia de enfoques anteriores [Tu91], [Mu92], [Ep94], [Pe94], [Na96b], [Oh96], [Le00], EP puede usar las imágenes determinadas por el algoritmo propuesto como su conjunto de imágenes de entrenamiento, en lugar del número mayor de ellas, actualmente usado. Más aún, EP no emplea splines para medir y controlar la precisión del modelo, sino como una herramienta para obtener una representación continua de la apariencia del objeto. *En contraste, aquí se emplea la interpolación lineal como medida de la precisión del modelo, y como criterio para seleccionar las imágenes del objeto estrictamente necesarias, para construir tal modelo para una precisión  $\epsilon$  dada.* Tanto medida como criterio son preservados al aplicar la técnica de espacios propios. Esto es cierto debido a que si  $\mathbf{i}'_t$  y  $\mathbf{i}'_u$  son dos imágenes pertenecientes a el conjunto de imágenes usado para calcular el espacio propio, y si  $\mathbf{f}_t$  y  $\mathbf{f}_u$  son las proyecciones de las respectivas imágenes, entonces el cuadrado de la distancia euclideana entre los puntos  $\mathbf{f}_t$  y  $\mathbf{f}_u$  es una aproximación a la SSD entre las imágenes  $\mathbf{i}'_t$  y  $\mathbf{i}'_u$ , es decir, la expresión:

$$\| \mathbf{i}'_t - \mathbf{i}'_u \|^2 \approx \| \mathbf{f}_t - \mathbf{f}_u \|^2 \quad (6)$$

se cumple [Na96a].

La expresión (6) implica que, por lo tanto, como el algoritmo propuesto usa SSD en el paso 8 como criterio para adquirir imágenes, el conjunto de imágenes

adquirido cumple con la restricción en la precisión  $\epsilon$ , y esta precisión es preservada por el modelo construido en el espacio propio.

# CAPÍTULO 5

## *GENERALIZACIÓN PARA N-PARÁMETROS*

**Resumen:** En este Capítulo se plantea la extensión de la propuesta hecha en el Capítulo previo, para considerar la apariencia de un objeto definida por múltiples parámetros, y obtener una representación multiparamétrica del objeto.

---

En el capítulo anterior se planteó la técnica para realizar la adquisición adaptativa de imágenes, para capturar la apariencia de un objeto. En esta técnica se supuso que el objeto se encuentra situado en el centro de la tornamesa, y que gira frente a la cámara; más aún, se supuso que la cámara se encontraba situada en una posición fija respecto a la tornamesa, y que no podía elevarse y/o descender sobre el plano de la tornamesa. Finalmente, también se supuso que la posición de la fuente de iluminación era también fija, con respecto a la tornamesa y a la cámara. Bajo las suposiciones anteriores, se mencionó que la apariencia del objeto describía una trayectoria  $\sigma: [0, 2\pi] \rightarrow \mathbf{R}^m$  parametrizada por el ángulo de rotación  $\theta$  del objeto frente a la cámara, respecto a una posición inicial fija (Ver Sección 2.2). Ahí el ángulo  $\theta$  se empleó para considerar todos los posibles valores que podía adoptar el parámetro, definido por el giro de la tornamesa frente al sensor.

Sin embargo, nótese que si se remueven algunas ó todas las suposiciones anteriores, entonces la tarea de capturar la apariencia del objeto deberá estar definida por más de un parámetro. Por ejemplo, si se asume que la cámara puede elevarse y/o descender, esto definirá el segundo parámetro para la apariencia del objeto; aunado a esto, si se asume que la fuente de iluminación puede elevarse y/o descender paralelamente al plano de la tornamesa (a partir de una posición inicial fija), se estará definiendo el tercer parámetro, y de esta manera, se pueden incrementar los parámetros de la tarea de visión. En general, algunos de los parámetros que se pueden considerar, incluyen a (sin estar restringido sólo a ellos):

- Ángulo de giro de la tornamesa
- Ángulo de ascenso/descenso de la cámara
- Posición de la fuente de iluminación, respecto al objeto (lo cual puede definir más de un parámetro)
- Posiciones relativas entre las partes que formen a un objeto articulado

Nótese entonces que si se define más de un parámetro para determinar la apariencia del objeto, entonces la técnica propuesta en el capítulo anterior deja de ser



aplicable, dado que fue propuesta para actuar solamente sobre un parámetro (definido por el giro del objeto frente al sensor).

En este capítulo se expone de qué manera pueden extenderse las ideas planteadas en el capítulo anterior, para analizar la apariencia de un objeto, definida ésta por  $N$  parámetros.

## 5.1 Adquisición de imágenes

Para crear modelos basados en apariencia, se comienza con la adquisición de imágenes del objeto a modelar, según se describió en la Sección 2.2. Sin embargo, a diferencia de lo mencionado en la Sección 2.2, para el caso en que la apariencia está determinada por más de un parámetro, se deben adquirir imágenes en posiciones específicas para cada uno de los posibles valores que pueda adoptar cada parámetro.

Concretamente, si la apariencia está determinada por  $N$  parámetros  $\theta_i$ ,  $i=1,\dots,N$ , se deberán adquirir imágenes en todos los puntos de la forma:  $(\theta_1, \theta_2, \dots, \theta_N)$ , donde los parámetros toman valores dentro de algún dominio  $D \subset \mathbf{R}^N$ . En cada uno de estos puntos se obtienen imágenes del objeto que si son representadas como matrices de  $n \times m$ , y las columnas de cada matriz son empiladas para formar vectores en  $\mathbf{R}^{nm}$  (como se mencionó en la Sección 1.2), se puede observar de la Definición 6, que tales vectores pertenecen a una superficie  $\sigma: D \rightarrow \mathbf{R}^{nm}$  definida por los parámetros que se emplean para modelar al objeto. Esta superficie es conocida en la literatura como **la superficie determinada por la apariencia del objeto**, o simplemente, **la apariencia del objeto** definida por  $N$  parámetros.

Al igual que en el caso de un parámetro, usualmente la apariencia del objeto definida por  $N$  parámetros, se discretiza al aplicar un muestreo uniforme sobre la región en que estén definidos los parámetros (ver por ejemplo, [Na96a]).

Evidentemente, y al igual que en el caso de un parámetro, emplear un muestreo uniforme no garantiza precisión alguna sobre el modelo generado, y para el caso de objetos con apariencia simple, (como el caso de una esfera pintada homogéneamente de un color), la cantidad de imágenes requerida será extremadamente grande, tal y como se mostró en la Tabla 1, Sección 3.1.

## 5.2 Interpolación N-lineal

Una alternativa eficiente de adquirir imágenes para obtener la apariencia de un objeto, que es parametrizada por  $N$  parámetros, la aporta la técnica propuesta en el Capítulo 4, si consideramos el concepto de interpolación N-lineal. Este concepto generaliza a la interpolación lineal, la cual interpola mediante el empleo de una línea recta (la curva más simple), los puntos comprendidos entre los puntos  $\mathbf{b}_0$  y  $\mathbf{b}_1$ ,

mediante la expresión (5). Por supuesto, debido a que se están considerando  $N$  parámetros, se necesita aproximar una superficie parametrizada en lugar de una trayectoria (como fue expuesto en la Sección 5.1). Por esta razón se propone aproximar la superficie parametrizada con interpolación  $N$ -lineal, en lugar de interpolación lineal. La interpolación  $N$ -lineal es la generalización "natural" de la interpolación lineal, debido a que si  $N=1$ , se obtiene una interpolación lineal (interpolación 1-lineal).

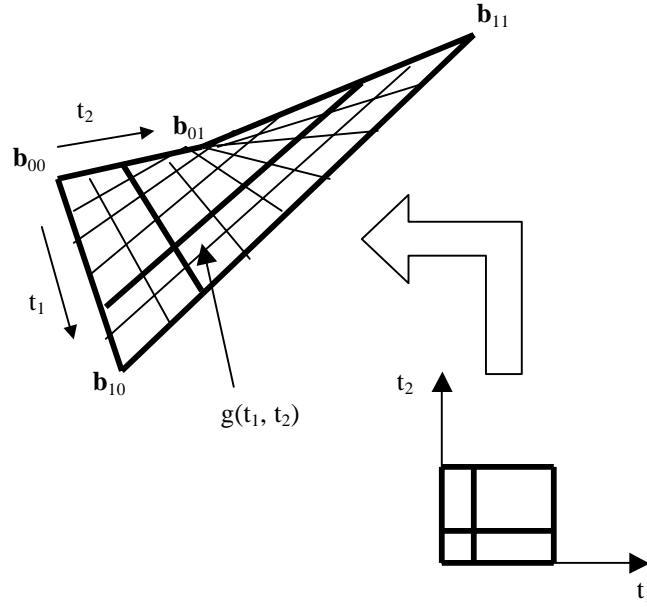


Figura 13: La interpolación bilineal interpola entre 4 puntos, por medio de la superficie más simple entre ellos: un paraboloide hiperbólico (adaptado de [Fa88]).

Para el caso bidimensional, se interpola entre 4 puntos dados mediante la superficie más simple entre ellos (un paraboloide hiperbólico), obtenida mediante la interpolación bilineal de los puntos  $\mathbf{b}_{00}$ ,  $\mathbf{b}_{01}$ ,  $\mathbf{b}_{10}$  y  $\mathbf{b}_{11}$ , dada por las siguientes expresiones [Fa88]:

$$\begin{aligned}
 \mathbf{B}_{00}^{10} &= (1-t_1)\mathbf{b}_{00} + t_1\mathbf{b}_{10} & ; & & 0 \leq t_1 \leq 1 \\
 \mathbf{B}_{01}^{11} &= (1-t_1)\mathbf{b}_{01} + t_1\mathbf{b}_{11} & ; & & 0 \leq t_1 \leq 1 \\
 g(t_1, t_2) &= (1-t_2)\mathbf{B}_{00}^{10} + t_2\mathbf{B}_{01}^{11} & & & 0 \leq t_1 \leq 1; 0 \leq t_2 \leq 1
 \end{aligned} \tag{7}$$

donde los términos:  $\mathbf{B}_{00}^{10}$  y  $\mathbf{B}_{01}^{11}$ , se emplean como paso intermedio para obtener el interpolante bilineal  $g(t_1, t_2)$ , de una manera sistemática, empleando únicamente interpoladores lineales. De hecho, nótese que  $\mathbf{B}_{00}^{10}$  y  $\mathbf{B}_{01}^{11}$ , son interpoladores lineales (según la expresión 5, Sección 4.2), sobre 2 de los lados de la región a interpolar, y

actúan únicamente sobre una dimensión (Ver Figura 13). Finalmente,  $g(t_1, t_2)$  interpola en ambas direcciones, al incluir a  $\mathbf{B}_{00}^{10}$  y  $\mathbf{B}_{01}^{11}$ , en su definición.

Un esquema de esta forma de interpolar, usando las expresiones (7), es mostrado en la Figura 13. La interpolación bilineal puede ser vista como un mapeo del cuadrado unitario  $0 \leq t_1, t_2 \leq 1$  sobre la superficie  $g(t_1, t_2)$ . El interpolante bilineal es lineal en ambas direcciones  $t_1$  y  $t_2$ .

Para el caso tridimensional, se interpola entre 8 puntos dados  $\mathbf{b}_{000}$ ,  $\mathbf{b}_{100}$ ,  $\mathbf{b}_{010}$ ,  $\mathbf{b}_{110}$ ,  $\mathbf{b}_{001}$ ,  $\mathbf{b}_{101}$ ,  $\mathbf{b}_{011}$  y  $\mathbf{b}_{111}$ , mediante el volumen más simple entre ellos (que no posee una representación física), obtenida mediante la interpolación trilineal de los 8 puntos. Tal forma de interpolación es obtenida mediante el empleo de las siguientes expresiones:

$$\begin{aligned}
 \mathbf{B}_{000}^{100} &= (1-t_1)\mathbf{b}_{000} + t_1\mathbf{b}_{100} & ; & & 0 \leq t_1 \leq 1 \\
 \mathbf{B}_{010}^{110} &= (1-t_1)\mathbf{b}_{010} + t_1\mathbf{b}_{110} & ; & & 0 \leq t_1 \leq 1 \\
 \mathbf{B}_{001}^{101} &= (1-t_1)\mathbf{b}_{001} + t_1\mathbf{b}_{101} & ; & & 0 \leq t_1 \leq 1 \\
 \mathbf{B}_{011}^{111} &= (1-t_1)\mathbf{b}_{011} + t_1\mathbf{b}_{111} & ; & & 0 \leq t_1 \leq 1 \\
 \\ 
 \mathbf{B}_{000}^{110} &= (1-t_2)\mathbf{B}_{000}^{100} + t_2\mathbf{B}_{010}^{110} & ; & & 0 \leq t_2 \leq 1 \\
 \mathbf{B}_{001}^{111} &= (1-t_2)\mathbf{B}_{001}^{101} + t_2\mathbf{B}_{011}^{111} & ; & & 0 \leq t_2 \leq 1 \\
 \\ 
 g(t_1, t_2, t_3) &= (1-t_3)\mathbf{B}_{000}^{110} + t_3\mathbf{B}_{001}^{111} & & & 0 \leq t_1 \leq 1; 0 \leq t_2 \leq 1; 0 \leq t_3 \leq 1 & \quad (8)
 \end{aligned}$$

En este caso, la interpolación trilineal puede ser vista como un mapeo del cubo unitario  $0 \leq t_1, t_2, t_3 \leq 1$ , sobre el volumen  $g(t_1, t_2, t_3)$ . El interpolante trilineal es lineal en cada una de las direcciones  $t_1, t_2, t_3$ . Como puede observarse de (7) y (8), y según se cita en [Fa88], es posible obtener las expresiones generales para realizar una interpolación N-lineal entre  $2^N$  puntos (esto es, la interpolación  $g(t_1, t_2, \dots, t_N)$  que involucra N parámetros), mediante un procedimiento simple y algorítmico, aún cuando aquí no se expondrá éste.

### 5.3 Adquisición adaptativa de imágenes generalizada

Como se puede observar de la Sección previa, la interpolación N-lineal puede ser usada para aproximar superficies parametrizadas dentro de un error  $\epsilon$ , de acuerdo a la Definición 7. De esta manera, tal interpolación puede ser usada en la técnica básica propuesta en el capítulo 4, si las siguientes pautas son consideradas:

- 1) El espacio de trabajo debe ser cambiado para considerar más de un parámetro.
- 2) La técnica debe ser cambiada para usar alguna de las expresiones (5), (7), (8) o derivar las apropiadas (como se explicó previamente) si es necesario el uso de más de 3 parámetros.
- 3) La técnica básica debe ser cambiada para dividir la zona de análisis en  $2^N$  sub-zonas para analizar cada una por separado: con un parámetro, se divide el intervalo en 2 sub-intervalos; con 2 parámetros, se divide un cuadrado en cuatro cuadrados, etc., usando el centro de cada zona como referencia para la división.
- 4) Las estructuras de datos deben ser cambiadas para soportar más de un parámetro.

Para aclarar las ideas discutidas previamente, en la siguiente sección se expondrá el algoritmo generalizado para N-parámetros, para determinar las imágenes estrictamente necesarias, para construir el modelo de un objeto. Se presentará una versión compacta de tal algoritmo, reservando los detalles del algoritmo en extenso, para referir al lector al apéndice C.

#### **Algoritmo 2.**

- 0.- Hacer  $\epsilon$  igual a la precisión deseada.**
- 1.- Colocar los límites inferiores para cada uno de los N parámetros.**
- 2.- Colocar los límites superiores para cada uno de los N parámetros.**
- 3.- Adquirir  $2^N$  imágenes, una para cada vértice de la zona a analizar.**
- 4.- Si se ha rebasado la precisión máxima del hardware, hacer Error = 1 e ir a 14.**
- 5.- Calcular el punto medio de la zona a analizar ( $L_m$ ).**
- 6.- Adquirir imagen en la posición  $L_m$ .**
- 7.- Interpolarse N-linealmente entre las  $2^N$  imágenes adquiridas en el paso 3, usando alguna de las expresiones (5), (7), (8), o la expresión apropiada para el caso, para obtener  $g(t_1, t_2, \dots, t_N)$ ,  $0 \leq t_i \leq 1$ .**
- 8.- Si es posible interpolar la imagen  $L_m$  con  $g(0.5, 0.5, \dots, 0.5)$  dentro de un error  $\epsilon$ , usando la suma del cuadrado de las diferencias (SSD) como criterio para comparar  $L_m$  y  $g(0.5, 0.5, \dots, 0.5)$ , esto es,**  

$$\| L_m - g(0.5, 0.5, \dots, 0.5) \|^2 \leq \epsilon$$
**entonces ir al paso 11.**
- 9.- Dividir la zona en  $2^N$  sub-zonas. Almacenar en la pila las  $2^N$  sub-zonas.**
- 10.- Ir a 12.**
- 11.- Almacenar las imágenes adquiridas en el paso 3, como imágenes necesarias para construir el modelo del objeto.**
- 12.- Si la pila no está vacía, desempilar una zona a analizar e ir al paso 4.**
- 13.- Error = 0;**
- 14.- Fin.**

## **5.4 Algoritmo generalizado para N-parámetros**

En el Algoritmo 2 se muestra el algoritmo adaptativo generalizado, para N parámetros. Este algoritmo generalizado adquiere las imágenes estrictamente necesarias para construir el modelo de un objeto, cuando la tarea de adquirir las imágenes está definida por N parámetros. El algoritmo está basado en un esquema de interpolación N-lineal y SSD para decidir que imágenes son estrictamente necesarias para construir el modelo del objeto.

## **5.5 Consideraciones sobre el algoritmo generalizado propuesto**

Un aspecto importante que debe ser considerado se refiere al paso 8 del algoritmo presentado. Aquí, la división puede ser hecha homogéneamente o heterogéneamente. La división homogénea es deseable si todos los parámetros son definidos inicialmente con los mismos límites y los sistemas mecánicos requeridos para adquirir las imágenes tienen la misma resolución para todos los parámetros. Si este no es el caso, se debe preferir la división heterogénea. En este caso, después de determinar que se requiere una división, se debe verificar si tal división deberá hacerse sobre todas las N direcciones que definen a la superficie. Esto puede ser realizado mediante N interpolaciones lineales, una para cada parámetro. El sistema de software que se implementó para experimentar tales ideas y que será expuesto en el siguiente capítulo, emplea división heterogénea. Nótese que la división homogénea es un caso particular incluido en la división heterogénea.

# CAPÍTULO 6

## *RESULTADOS EXPERIMENTALES*

**Resumen:** En este Capítulo se exponen resultados experimentales obtenidos al implementar las ideas planteadas en los Capítulos 4 y 5, en un sistema de software que fue empleado para analizar 2 bases de imágenes de objetos. Así mismo, es presentado un análisis de los resultados obtenidos.

---

Fue desarrollado un sistema de software para adquirir las imágenes necesarias para modelar la apariencia de objetos, con las propuestas presentadas en los Capítulos 4 y 5 (un parámetro y múltiples parámetros, respectivamente). Este sistema de software fue acoplado al sistema de hardware para la adquisición de imágenes, que fue descrito en el Capítulo 1, en la Figura 3. El sistema de adquisición de imágenes recibió los comandos del sistema de software para adquirir imágenes en las posiciones determinadas por el algoritmo. El sistema, en su conjunto, determinó cuántas y cuáles imágenes fueron requeridas para satisfacer el criterio de precisión  $\epsilon$ , para cada objeto estudiado.

En el presente Capítulo se describen en extenso los resultados experimentales obtenidos con la propuesta realizada en los Capítulos 4 y 5. Estos resultados fueron presentados también en sus versiones preliminares en [AI02a] y [AI02b].

### **6.1 Un parámetro**

En esta sección se exponen los principales resultados experimentales obtenidos con la técnica básica, expuesta en el Capítulo 4. Para obtener estos resultados se empleó únicamente un parámetro para determinar la apariencia de los objetos; este parámetro quedó determinado por el ángulo de giro de la tornamesa, del sistema de hardware para la adquisición de imágenes.

#### **6.1.1 Objetos de estudio**

El algoritmo 1, expuesto en el Capítulo 4, fue aplicado a 2 conjuntos de objetos. El primero de ellos es un conjunto de objetos propios, y es mostrado en la Figura 14. El segundo, mostrado en la Figura 15, es el conjunto de imágenes contenido en COIL-20 (*Columbia Object Image Library*), y pertenece a la Universidad de Columbia [Na94b]. Este último conjunto de imágenes fue empleado, debido a que COIL-20 es una referencia clásica y extremadamente empleada en la

literatura, para probar la eficiencia de los sistemas propuestos (Ver por ejemplo, [Sc97], [Sc00], [Le00]). COIL-20 está formado por un subconjunto de 20 objetos de un total de 100 que integran a COIL, la base de imágenes completa, perteneciente a la Universidad de Columbia.

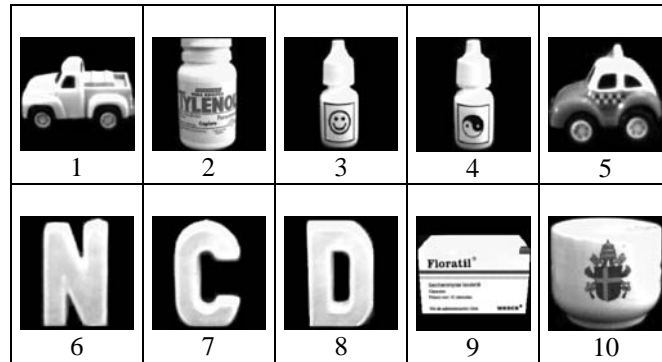


Figura 14: Algunos objetos propios empleados para experimentar el algoritmo 1.

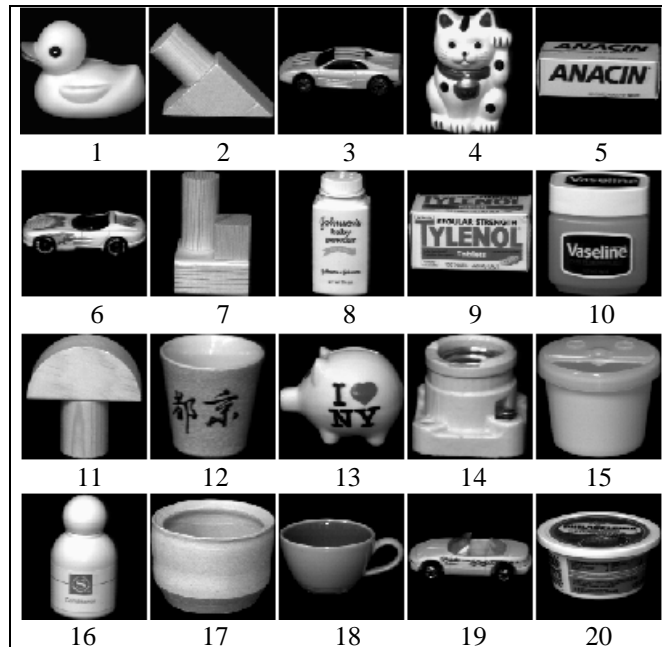


Figura 15: Objetos que forman a COIL-20<sup>4</sup> (Columbia Object Image Library).

<sup>4</sup> Imágenes tomadas de la base de imágenes COIL (Columbia Object Image Library), creada por el Dept. of Computer Science, Columbia University [Na96b].

Los objetos de estudio empleados, están formados por una amplia variedad de materiales (plástico, cartón, cerámica, madera, metal, etc.) con diferentes propiedades de reflexión. Así mismo, presentan una amplia gama de propiedades geométricas, tanto simples (objetos cilíndricos) hasta complejas (piezas de madera, autos, objetos de uso cotidiano). También incluyen una amplia variedad de grabados y colores. Esta amplia gama de características, hace que se tenga representado un amplio espectro de apariencias de objetos.

### 6.1.2 Determinación del número de imágenes requeridas

Como una primera etapa, se procedió a aplicar el algoritmo propuesto sobre los objetos mostrados en las Figuras 14 y 15, con diversos valores para la precisión deseada  $\epsilon$ . Los resultados del experimento son mostrados en las Tablas 2 y 3, para los objetos mostrados en las Figuras 14 y 15, respectivamente. En ellas, las celdas etiquetadas con + indican que para alcanzar la precisión requerida  $\epsilon$ , se necesitaba que el incremento en el ángulo de giro de la tornamesa entre imágenes sucesivas fuera menor a  $5^\circ$ . Sin embargo, para este experimento, se limitó al sistema de rotación de los objetos a no superar esta precisión (en la literatura típicamente se reporta una precisión de entre  $12^\circ$  a  $10^\circ$  como máximo; ver, por ejemplo, [Na96a]). Aún cuando en las Tablas 2 y 3 no se muestra, el algoritmo determinó, adicionalmente al número de imágenes necesarias, cuales son las imágenes estrictamente necesarias para cada  $\epsilon$  (ver la siguiente Sección).

		OBJETO									
		1	2	3	4	5	6	7	8	9	10
	$\epsilon = 2000$	+	+	+	+	+	+	+	+	+	+
	$\epsilon = 2500$	+	9	3	3	+	+	+	+	+	+
<b>P</b>	$\epsilon = 3000$	+	3	2	2	+	+	+	+	+	+
<b>R</b>	$\epsilon = 3500$	+	2	2	2	+	30	20	29	+	4
<b>E</b>	$\epsilon = 4000$	+	2	2	2	+	21	17	21	+	2
<b>C</b>	$\epsilon = 4500$	+	2	2	2	+	18	14	16	20	2
<b>I</b>	$\epsilon = 5000$	+	2	2	2	27	14	11	15	18	2
<b>S</b>	$\epsilon = 5500$	+	2	2	2	20	12	10	11	14	2
<b>I</b>	$\epsilon = 6000$	+	2	2	2	18	10	9	10	14	2
<b>O</b>	$\epsilon = 6500$	23	2	2	2	14	10	9	10	12	2
<b>N</b>	$\epsilon = 7000$	21	2	2	2	12	8	7	9	10	2
	$\epsilon = 7500$	18	2	2	2	9	8	5	8	7	2
	$\epsilon = 8000$	15	2	2	2	7	8	5	5	5	2

Tabla 2: Número de imágenes requeridas para capturar la apariencia de los objetos mostrados en la Figura 14, dependiendo de distintos valores de  $\epsilon$ . Las celdas marcadas con + indican que para alcanzar la precisión  $\epsilon$  señalada, el ángulo de rotación de la tornamesa entre imágenes sucesivas debe ser inferior a  $5^\circ$ .



Analizando los resultados mostrados en ambas Tablas, se puede corroborar que, como fue mencionado en la introducción, la adquisición de imágenes depende de la precisión deseada  $\epsilon$  que el investigador desee, y de la clase de objeto a ser considerado. De esta forma, para lograr alcanzar una determinada precisión en el modelo, se requiere una cantidad distinta de imágenes para objetos diferentes. Observando los resultados obtenidos, se constata la siguiente propiedad importante: a mayor precisión exigida para el modelo ( $\epsilon$  pequeño), mayor cantidad de imágenes. Esta propiedad contrasta fuertemente con los resultados encontrados en la literatura, que suponen que una cantidad fija de imágenes, es suficiente para capturar la apariencia de cualquier objeto.

		OBJETO																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	$\epsilon = 1500$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	6	9	+	+	+	+
	$\epsilon = 2000$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	3	2	+	+	+	+
	$\epsilon = 2500$	27	+	+	+	+	+	29	23	+	+	+	7	+	+	2	2	+	+	+	+
<b>P</b>	$\epsilon = 3000$	23	+	+	26	+	+	20	17	+	+	25	2	+	+	2	2	5	22	+	+
<b>R</b>	$\epsilon = 3500$	17	+	+	24	+	+	16	13	+	21	20	2	+	26	2	2	3	15	+	+
<b>E</b>	$\epsilon = 4000$	14	24	+	21	+	+	13	9	+	17	15	2	+	22	2	2	2	10	+	8
<b>C</b>	$\epsilon = 4500$	13	19	+	13	+	+	10	8	+	15	15	2	19	2	2	2	2	7	+	4
<b>I</b>	$\epsilon = 5000$	9	17	+	13	+	+	8	6	+	10	12	2	17	2	2	2	2	5	18	3
<b>S</b>	$\epsilon = 5500$	9	13	18	11	22	+	8	5	+	8	9	2	12	2	2	2	2	2	16	2
<b>I</b>	$\epsilon = 6000$	8	13	18	10	20	+	5	4	26	6	8	2	9	2	2	2	2	2	14	2
<b>O</b>	$\epsilon = 6500$	8	12	14	7	16	+	5	4	22	5	7	2	9	2	2	2	2	2	11	2
<b>N</b>	$\epsilon = 7000$	7	11	14	6	14	+	4	2	20	4	6	2	6	2	2	2	2	2	11	2
	$\epsilon = 7500$	7	11	9	5	12	16	2	2	17	4	4	2	6	2	2	2	2	2	9	2
	$\epsilon = 8000$	5	8	6	4	11	15	2	2	14	4	4	2	6	2	2	2	2	2	6	2

Tabla 3: Número de imágenes requeridas para capturar la apariencia de los objetos en COIL-20 (Figura 15), dependiendo de distintos valores de  $\epsilon$ . Las celdas marcadas con + indican que para alcanzar la precisión  $\epsilon$  señalada, el ángulo de rotación de la tornamesa entre imágenes sucesivas debe ser inferior a  $5^\circ$ .

Otro hecho interesante que se puede observar de estos resultados, radica en que el rango de valores permitidos para la precisión  $\epsilon$  (esto es, el rango de valores que no rebasan la precisión del hardware), es propia de cada objeto en particular. Dado que el investigador no cuenta con ninguna información *a priori* sobre este rango, para cada objeto particular, surge la necesidad de que el sistema lo determine de manera automática. Este problema será abordado en el Capítulo 8.

### 6.1.3 Determinación de las imágenes estrictamente necesarias

Adicionalmente a la cantidad de imágenes requeridas para una precisión  $\epsilon$  dada, el algoritmo 1 es capaz de determinar cuáles son las imágenes estrictamente

necesarias para la precisión  $\epsilon$  deseada para el modelo. En las Tablas 4 y 5 se listan las imágenes estrictamente necesarias para alcanzar la precisión  $\epsilon$  señalada, para los objetos en las Figuras 14 y 15, respectivamente. Para poder entender los resultados contenidos en esas tablas, es necesario explicar cómo está estructurada COIL-20.

COIL-20 contiene imágenes de los 20 objetos mostrados en la Figura 15. Está formada por 72 imágenes por objeto (5 grados de incremento entre imágenes consecutivas), adquiridas como se explicó en la Sección 2.2. El conjunto total de imágenes está formado por 1440 imágenes (72 imágenes  $\times$  20 objetos). Las imágenes en COIL-20 están indexadas de 0 a 71, para cada objeto, y normalizadas en escala para ser de  $128 \times 128$  píxeles. De una manera similar se adquirieron y estructuraron las imágenes correspondientes a los objetos de la Figura 14 (720 imágenes: 72 imágenes  $\times$  10 objetos). Entonces, el ángulo de rotación de la tornamesa correspondiente a la  $k$ -ésima imagen es obtenido por multiplicar  $k \cdot 5^\circ$ ,  $0 \leq k \leq 71$ .

Numero de objeto	Precisión máxima ( $\epsilon$ ) lograda con 36 imágenes	Imágenes estrictamente necesarias para alcanzar la precisión máxima ( $\epsilon$ )	
			Total
1	6405	0, 8, 12, 14, 16, 18, 20, 22, 24, 26, 30, 36, 40, 44, 48, 50, 52, 54, 56, 58, 60, 62, 66	<b>23</b>
2	2307	0, 2, 4, 8, 12, 18, 26, 36, 44, 54, 62, 66	<b>12</b>
3	2715	0, 18, 36, 54	<b>4</b>
4	2671	0, 18, 36	<b>3</b>
5	4992	0, 4, 8, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 36, 40, 44, 48, 50, 52, 54, 58, 60, 62, 64, 66, 68	<b>27</b>
6	3295	0, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 36, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68	<b>32</b>
7	3138	0, 4, 8, 12, 14, 16, 18, 20, 22, 24, 26, 30, 36, 44, 46, 48, 50, 52, 54, 58, 60, 62, 64, 66	<b>24</b>
8	3395	0, 4, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 36, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 66, 68	<b>30</b>
9	4201	0, 4, 8, 10, 12, 14, 16, 18, 22, 24, 26, 28, 30, 32, 36, 44, 46, 48, 50, 54, 58, 60, 62, 64, 66, 68	<b>26</b>
10	3297	0, 8, 12, 14, 16, 18, 22, 26, 28, 30, 36, 40, 44, 48, 54, 62, 64, 66	<b>18</b>

Tabla 4: Análisis de la precisión de los objetos mostrados en la Figura 14.

Numero de objeto	Precisión máxima (ε) lograda con 36 imágenes de COIL-20	Imágenes estrictamente necesarias para alcanzar la precisión máxima (ε)	
			Total
1	2073	0, 4, 6, 8, 10, 12, 14, 16, 18, 22, 24, 26, 28, 30, 32, 34, 36, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68	32
2	3514	0, 4, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 36, 40, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 66	27
3	5103	0, 8, 12, 14, 16, 18, 20, 22, 26, 36, 44, 48, 50, 52, 54, 56, 58, 62	18
4	2945	0, 2, 4, 6, 8, 10, 12, 14, 18, 22, 26, 30, 36, 40, 44, 48, 50, 54, 56, 58, 60, 62, 64, 66, 68, 70	26
5	5252	0, 4, 8, 10, 12, 14, 16, 18, 20, 22, 26, 30, 36, 44, 48, 50, 52, 54, 56, 58, 62, 66, 68	23
6	7282	0, 8, 12, 14, 16, 18, 20, 22, 26, 36, 44, 48, 50, 52, 54, 56, 58, 62	18
7	2360	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 26, 30, 32, 34, 36, 38, 40, 42, 44, 48, 50, 52, 54, 56, 58, 62, 66, 68, 70	31
8	2152	0, 2, 4, 6, 8, 12, 18, 22, 26, 30, 32, 34, 36, 38, 40, 42, 44, 48, 50, 54, 58, 62, 66, 68, 70	25
9	5817	0, 4, 8, 12, 14, 16, 18, 20, 22, 24, 26, 30, 32, 36, 38, 40, 44, 48, 50, 52, 54, 56, 58, 62, 66, 68	26
10	3287	0, 2, 4, 6, 8, 12, 14, 18, 22, 26, 30, 32, 36, 40, 44, 48, 54, 58, 62, 66, 68, 70	22
11	2509	0, 4, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 40, 44, 48, 50, 52, 54, 56, 58, 60, 62, 66, 68	29
12	2319	0, 2, 4, 8, 18, 36, 54, 62, 66, 68, 70	11
13	4164	0, 2, 4, 6, 8, 10, 12, 18, 22, 26, 30, 36, 40, 44, 48, 50, 54, 58, 62, 66, 68, 70	22
14	3448	0, 2, 4, 8, 12, 14, 16, 18, 20, 22, 26, 30, 32, 36, 38, 40, 44, 48, 50, 54, 56, 58, 62, 66, 68, 70	26
15	1445	0, 18, 36, 44, 54, 62	6
16	1267	0, 2, 4, 6, 8, 18, 26, 36, 54, 62, 66, 68, 70	13
17	2733	0, 18, 26, 36, 54, 62	6
18	2545	0, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 54, 62, 64, 66, 68	26
19	4655	0, 8, 12, 14, 16, 18, 20, 22, 26, 36, 44, 48, 50, 52, 54, 56, 58, 62	18
20	3647	0, 8, 18, 26, 30, 36, 54, 62, 66	9

Tabla 5: Análisis de la precisión de COIL-20 (Figura 15).

Para poder tener un punto de comparación respecto a una adquisición de imágenes uniforme, se procedió a calcular la *precisión máxima* para cada objeto, de la siguiente manera: usualmente en trabajos previos se han usado las imágenes etiquetadas con un número par para construir los modelos de los objetos, y las imágenes con número impar son empleadas para probar los sistemas de reconocimiento. Entonces, se aplicó la expresión (5) a cada pareja de imágenes pares

consecutivas (esto es,  $\mathbf{I}_0\text{-}\mathbf{I}_2$ ,  $\mathbf{I}_2\text{-}\mathbf{I}_4$ , etc.), y se calculó  $g(0.5)$  para cada pareja. Enseguida se empleó SSD para comparar  $g(0.5)$  con la imagen impar comprendida entre la pareja de imágenes pares. Por ejemplo, para la pareja ( $\mathbf{I}_0$ ,  $\mathbf{I}_2$ ), se obtuvo  $g_{0,2}(0.5)$  mediante la expresión (5), y se comparó con  $\mathbf{I}_1$ , para obtener  $\delta_1 = \text{SSD}(g_{0,2}(0.5), \mathbf{I}_1)$ . Para la siguiente pareja ( $\mathbf{I}_2$ ,  $\mathbf{I}_4$ ), se obtuvo  $\delta_2$ , etc. Aplicando el mismo proceso a todas las parejas, se obtuvieron  $\delta_1, \delta_2, \dots, \delta_{36}$ . Enseguida se procedió a calcular la precisión máxima para cada objeto como  $\varepsilon = \max(\delta_1, \delta_2, \dots, \delta_{36})$ . Este valor implica que no es posible alcanzar una mejor precisión para el modelo del objeto en cuestión, usando las 36 imágenes pares solamente, porque existe al menos una pareja de imágenes cuya precisión para representar al objeto es de a lo más ese valor.

En las Tablas 4 y 5 se listan tanto el número de imágenes requeridas (columna 4), como los índices de las imágenes estrictamente necesarias para modelar a los objetos de las Figuras 14 y 15 (columna 3), empleando la máxima precisión  $\varepsilon$  (columna 2) para cada objeto en particular (columna 1). La raíz cuadrada de  $\varepsilon$  es mostrada en lugar de  $\varepsilon$ , por claridad.

De estos resultados experimentales, se puede constatar que el algoritmo 1 es capaz de determinar la complejidad de la apariencia de cada objeto, y en base a ello, adquirir la cantidad de imágenes apropiada para satisfacer los requerimientos de precisión  $\varepsilon$  que se le pidan. Esto puede constatarse de los resultados obtenidos para los objetos 2, 3 y 4 de la Figura 14, así como para los objetos 12, 15, 16, 17 y 20 de la Figura 15, los cuales presentan una apariencia simple, debida principalmente a su quasi-simetría rotacional, y en donde se adquirieron pocas imágenes. En contraste, para otros objetos fue necesario adquirir una mayor cantidad de imágenes.

#### **6.1.4 Distribución de las imágenes estrictamente necesarias**

Aún cuando los resultados mostrados en las Tablas 4 y 5 son significativos, no permiten realizar un análisis completo de los resultados obtenidos. Por esta razón, en las Figuras 16 y 17 se ha graficado la distribución de las imágenes obtenidas en la columna 3 de las Tablas 4 y 5, para los objetos de las Figuras 14 y 15, respectivamente, sobre el plano de la tornamesa del sistema para la adquisición de imágenes.

De estos resultados se puede observar cómo el algoritmo propuesto es capaz de localizar las zonas donde la apariencia varía poco y en consecuencia adquirir pocas imágenes; de una manera similar, es capaz de identificar las zonas con grandes variaciones en la apariencia y, en contraste, adquirir una gran cantidad de imágenes. En particular nótese cómo es capaz de localizar las etiquetas de los objetos 2, 3, 4 de la Figura 14, y 12 y 16 de la Figura 15.

Otro resultado importante que se puede observar de estas distribuciones, se refiere al efecto de la normalización de imágenes sobre los modelos de los objetos. Las distribuciones para los objetos 1, 5, 6, 7, 8 y 9 de la Figura 14, y 2, 3, 5, 6, 18 y 19 de la Figura 15, denotan la alta concentración de las imágenes adquiridas en las zonas donde se requiere un escalado considerable de las imágenes. Este efecto es observado en objetos que no presentan proporciones geométricas cercanas (esto es, el ancho es muy diferente del largo). Esto se debe a que el proceso de normalización induce considerables variaciones en la apariencia del objeto. Como consecuencia, si se emplea un muestreo uniforme el modelo así generado será altamente impreciso en esas zonas, aunque muy preciso en las restantes. Este efecto no ha sido considerado en la literatura existente y por tanto, esto repercutirá en posibles fallas de los sistemas de reconocimiento, si se pretende reconocer a los objetos en estas posiciones específicas, en especial si se requiere un reconocimiento entre objetos muy parecidos entre sí.

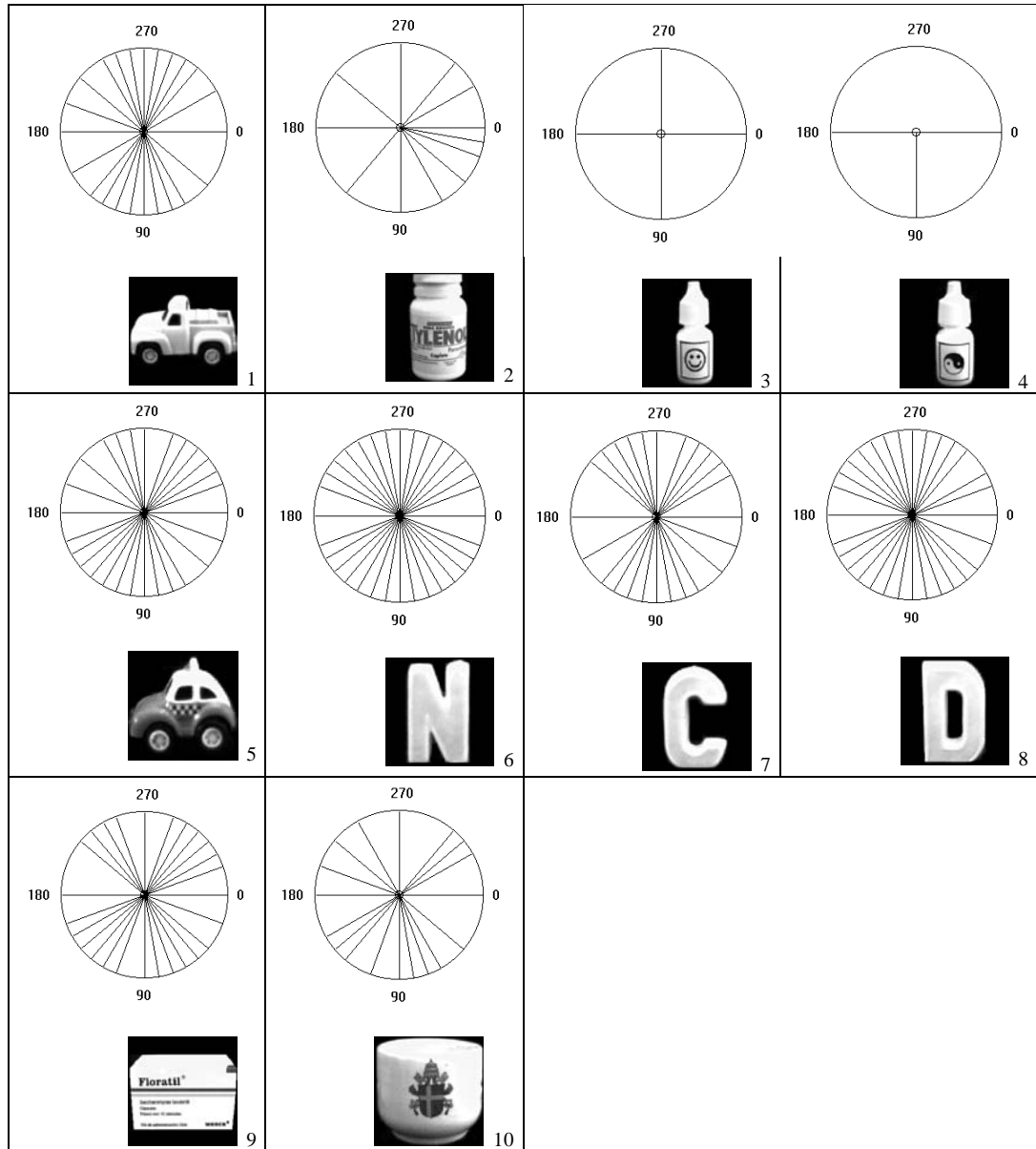


Figura 16: Distribución sobre el plano de la tornamesa de las imágenes estrictamente necesarias para modelar la apariencia de los objetos de la Figura 14.

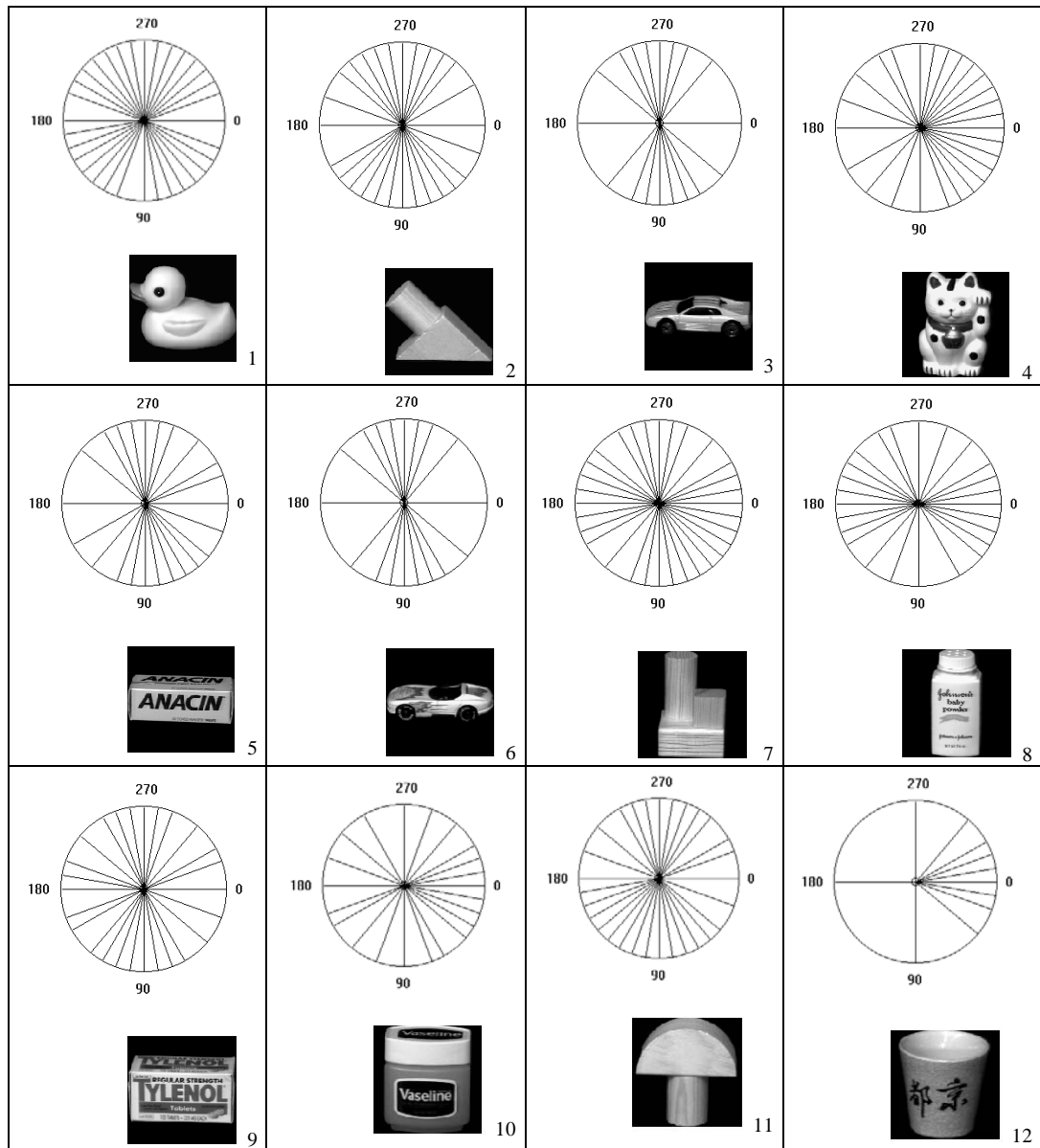


Figura 17: Graficación de la distribución de las imágenes estrictamente necesarias, para capturar la apariencia de los objetos que forman a COIL-20 (Figura 15).

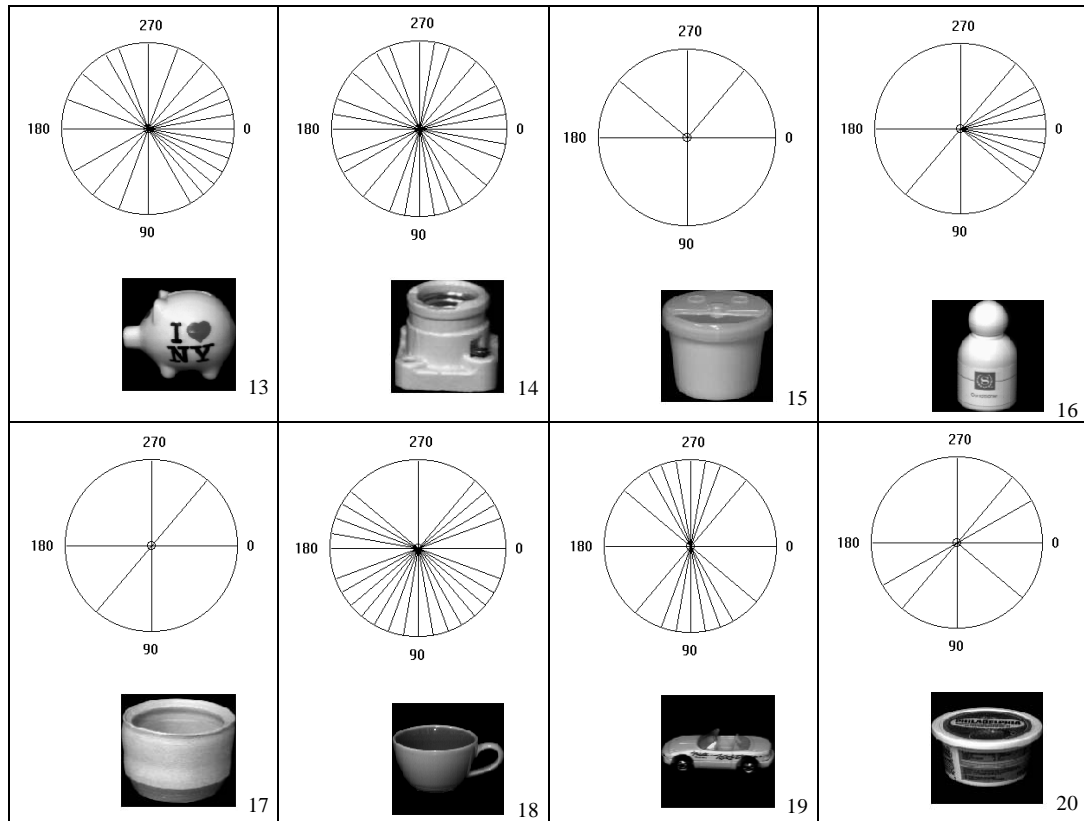


Figura 17(continuación): Graficación de la distribución de las imágenes estrictamente necesarias, para capturar la apariencia de los objetos que forman a COIL-20 (Figura 15).

### 6.1.5 Precisión de los modelos

Como una etapa mas en los resultados experimentales, se procedió a analizar la precisión de los modelos obtenidos con la técnica propuesta. Un modelo será preciso si es capaz de representar la apariencia del objeto para cualquier valor que adopte el parámetro que define a la apariencia, - dentro del rango permitido -, con una precisión  $\epsilon$ . En nuestro caso, el parámetro es el ángulo de giro de la tornamesa, y podrá adoptar cualquier valor  $0^\circ \leq \theta \leq 360^\circ$ . Aunque el parámetro puede adoptar, en teoría, cualquier valor real dentro del rango permitido, en la práctica es imposible permitir esto, dado que se requeriría una cantidad infinita de imágenes, para probar la precisión del modelo. Por esta razón en la práctica se prueba tal precisión, considerando una cantidad finita de puntos discretos, usualmente, espaciados igualmente entre sí. La precisión  $\epsilon$  será la que asigne el investigador.



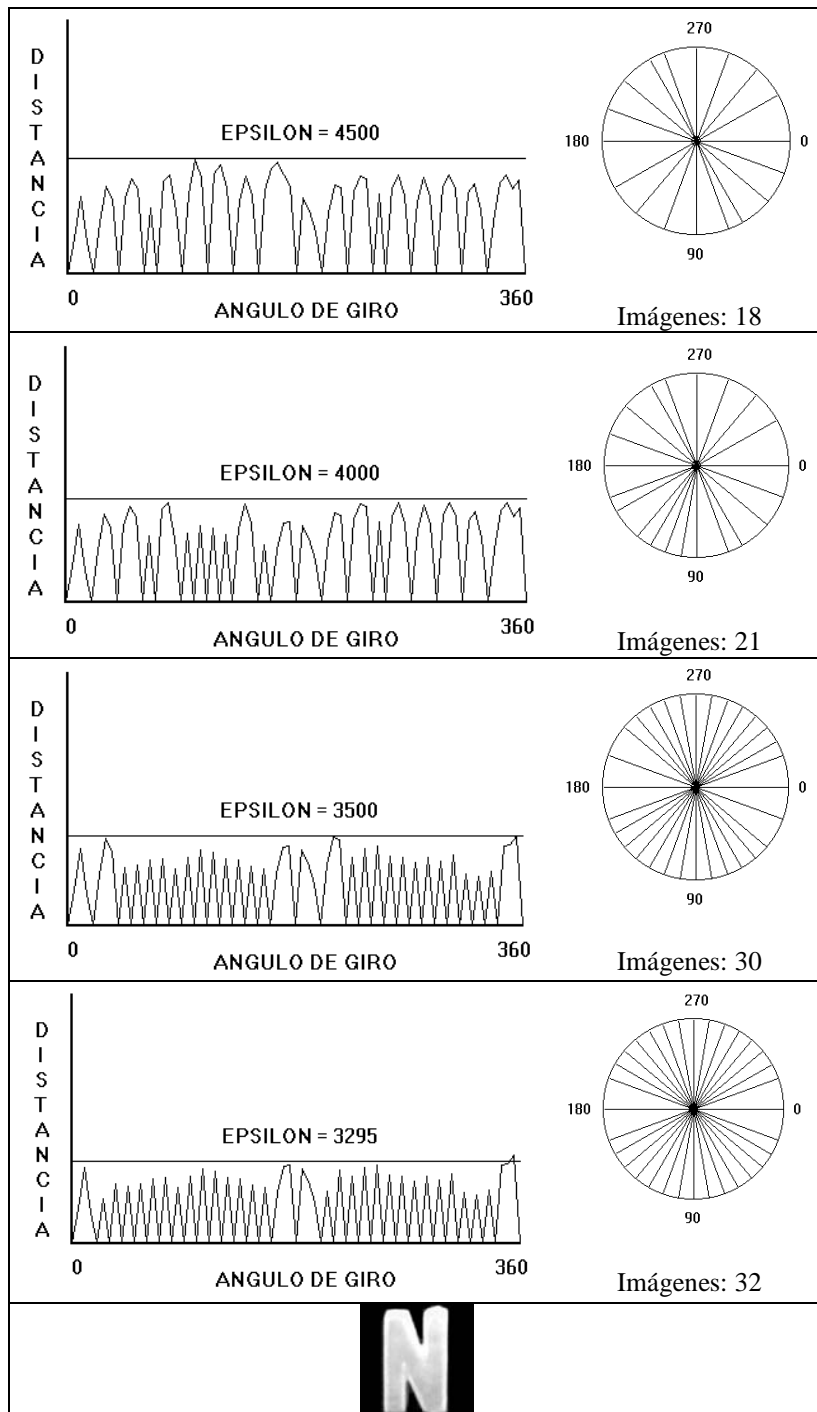


Figura 18: Análisis de la precisión de distintos modelos, generados mediante la técnica propuesta, para el objeto 6 de la Figura 14, en función de distintos valores para la precisión  $\epsilon$ . Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

Considerando lo anterior, se procedió a probar la precisión de los modelos obtenidos mediante la técnica propuesta, discretizando el modelo cada  $5^\circ$  de incremento del parámetro, para obtener un total de 72 imágenes provenientes del modelo. Paralelamente, se adquirieron 72 imágenes igualmente espaciadas ( $5^\circ$  de incremento), empleando el sistema de hardware para la adquisición de imágenes, ilustrado en la Figura 3. Enseguida se procedió a calcular la SSD entre las imágenes obtenidas mediante el modelo y las imágenes reales correspondientes. Finalmente, las distancias obtenidas mediante este procedimiento fueron graficadas, para obtener una interpretación gráfica del comportamiento del modelo, respecto a la apariencia del objeto.

En la Figura 18 se ilustran los resultados obtenidos al aplicar esta prueba de precisión al objeto 6 de la Figura 14, para diferentes valores para la precisión  $\epsilon$ . El eje X representa el ángulo de giro de la tornamesa, del sistema de hardware para la adquisición de imágenes. El eje Y grafica la SSD entre el modelo obtenido mediante la técnica propuesta, y la apariencia del objeto; para propósitos ilustrativos, se ha empleado la raíz cuadrada de la SSD, en lugar de la SSD, y la raíz cuadrada de  $\epsilon$ , en lugar de  $\epsilon$ . Así mismo, se muestra la distribución de las imágenes requeridas para cada caso, como se explicó en la Sección previa, junto con la cantidad de imágenes requeridas para construir el modelo.

En la Figura 18, las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones del modelo a la apariencia del objeto. Las partes sobre la precisión  $\epsilon$  indican que se debe mejorar el modelo en esas partes; dependiendo del grado de elevación del gráfico sobre la precisión  $\epsilon$  será el grado de imprecisión del modelo. Del análisis de esta Figura, se puede observar que la técnica propuesta es capaz de generar un modelo, que cumple con los requerimientos de precisión que asigne el investigador, tal y como se explicó en Capítulos previos, y que para cada precisión requerida adquiere un número apropiado de imágenes, en función de la precisión dada: a mayor precisión, mayor cantidad de imágenes requeridas. Nótese, sin embargo, que si se ha calculado el modelo para una determinada precisión  $\epsilon$  y se desea incrementar esta precisión a  $\epsilon'$ , con  $\epsilon > \epsilon'$ , basta adquirir algunas imágenes más en las zonas que el algoritmo determine, y adicionarlas al conjunto previamente determinado para la precisión  $\epsilon$ , por lo que el tiempo invertido previamente no es desperdiciado. Además, el no tener que analizar nuevamente el objeto desde el principio, es especialmente útil en la técnica de espacios propios [Mu95], en donde, con propuestas como la de Hall, Marshall y Martín [Ha00], es posible calcular un nuevo espacio propio de uno previamente calculado, al cual se le adicionan nuevas imágenes, en un tiempo muy reducido. Recuérdese que calcular un espacio propio requiere un cómputo intenso.

Para cada caso en la Figura 18, se puede observar que el modelo es extremadamente preciso, pues cumple con los requerimientos de precisión que se le exigieron (el gráfico está por debajo de la precisión  $\varepsilon$  señalada). Una excepción se presenta en el gráfico del 4o. renglón ( $\varepsilon = 3295$ ), en donde existe una zona en donde el gráfico rebasa la precisión requerida. Esto ocurre debido a que, como se explicó en el Capítulo 4, la técnica propuesta verifica solamente el punto medio de cada intervalo que analiza, para ajustar el modelo en ese punto, para alcanzar la precisión requerida. Puede observarse en el gráfico, que justamente el punto medio de ese intervalo coincide con la precisión exigida, sin embargo, algunos puntos distintos del punto medio en ese intervalo, no cumplen con la precisión exigida.

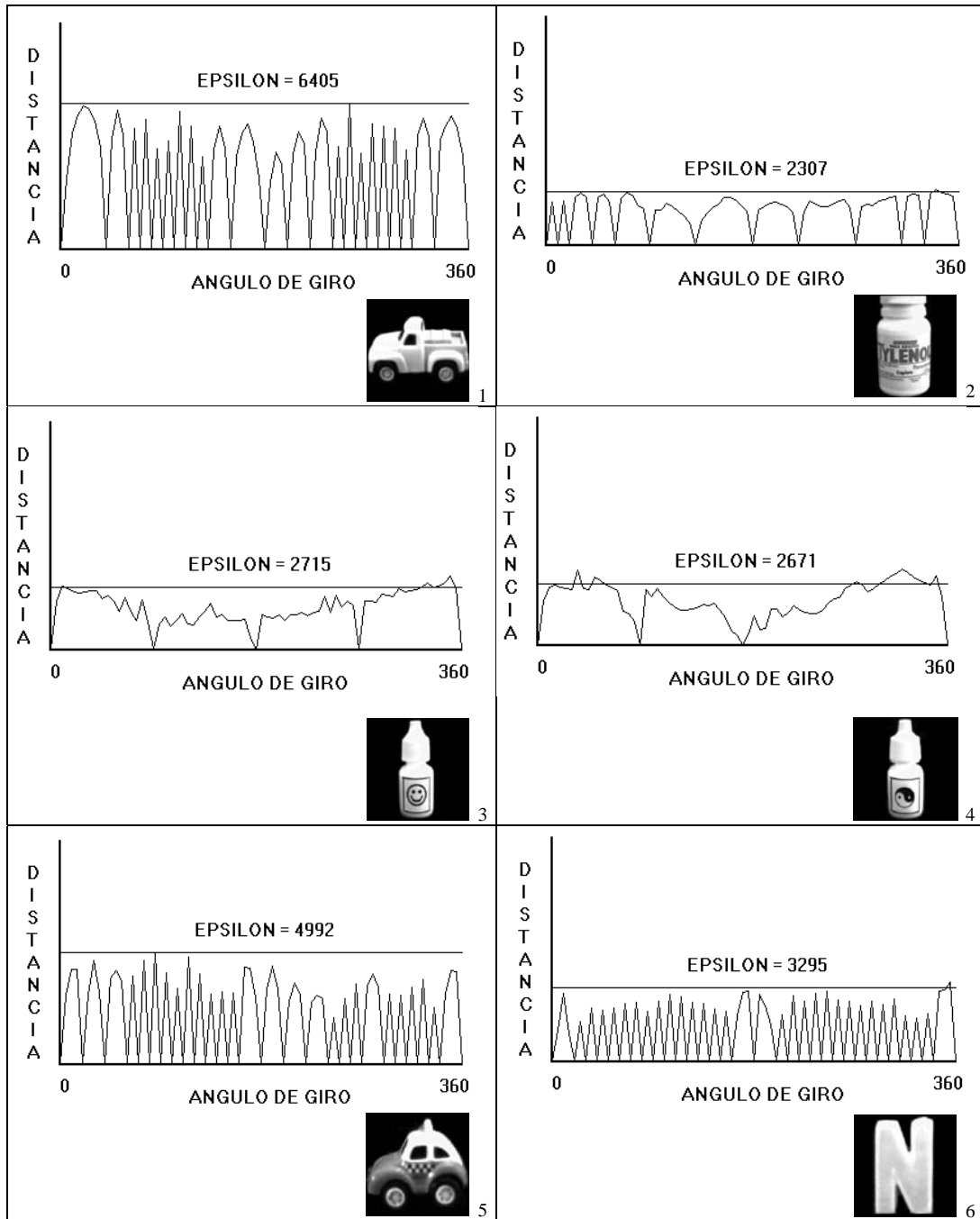


Figura 19: Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de la Figura 14, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

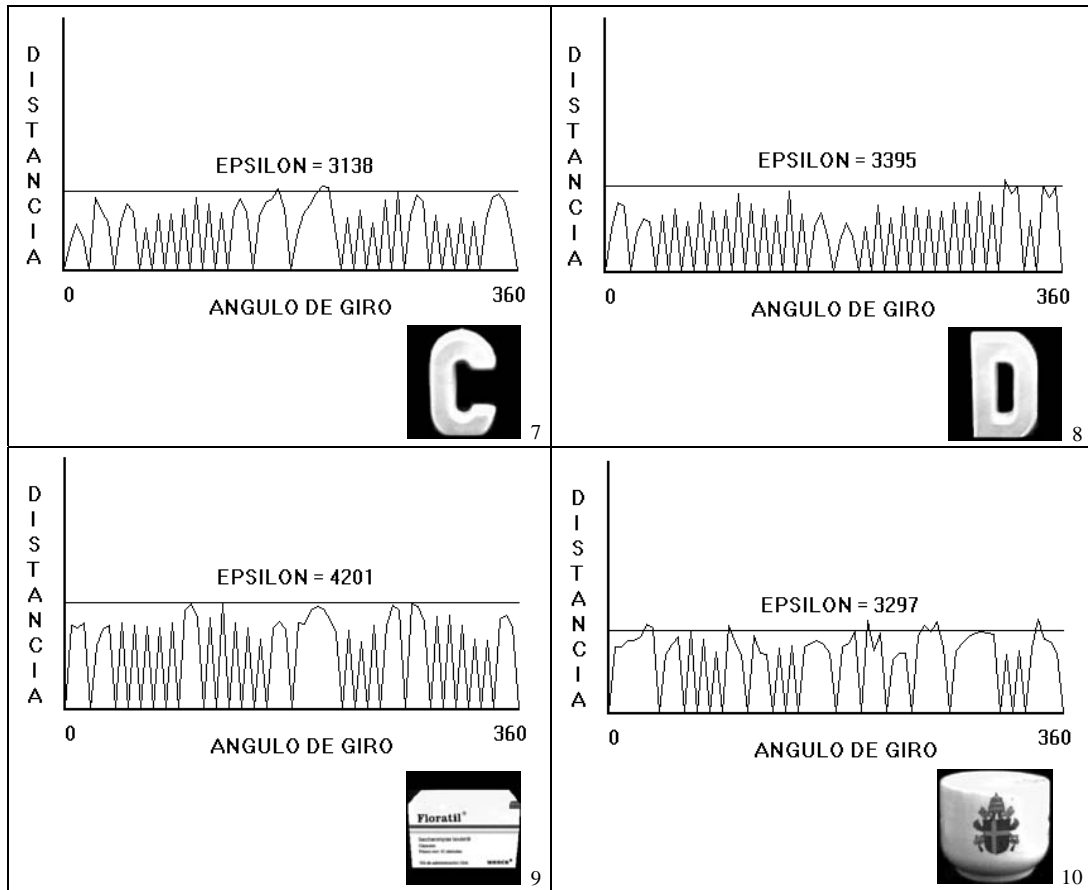


Figura 19(continuación): Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de la Figura 14, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

Sin embargo, puede notarse que el grado de desviación es muy pequeño, por lo que aún en este caso, el modelo es una buena aproximación a la apariencia del objeto. Nótese que este problema se presenta en una única zona, y que el resto del modelo es una buena aproximación. Aún cuando este problema es razonablemente tolerable para propósitos de reconocimiento de objetos, debido a que el grado de desviación es muy pequeño, en el Capítulo 8 se expondrá una manera de corregir este problema en la técnica propuesta.

En las Figuras 19 y 20 se muestra un análisis similar al de la Figura 18, de la precisión de los modelos generados con la técnica propuesta, para los objetos de las Figuras 14 y 15, empleando la precisión máxima para cada objeto, dada en las Tablas 4 y 5 (Sección 6.1.3), respectivamente.

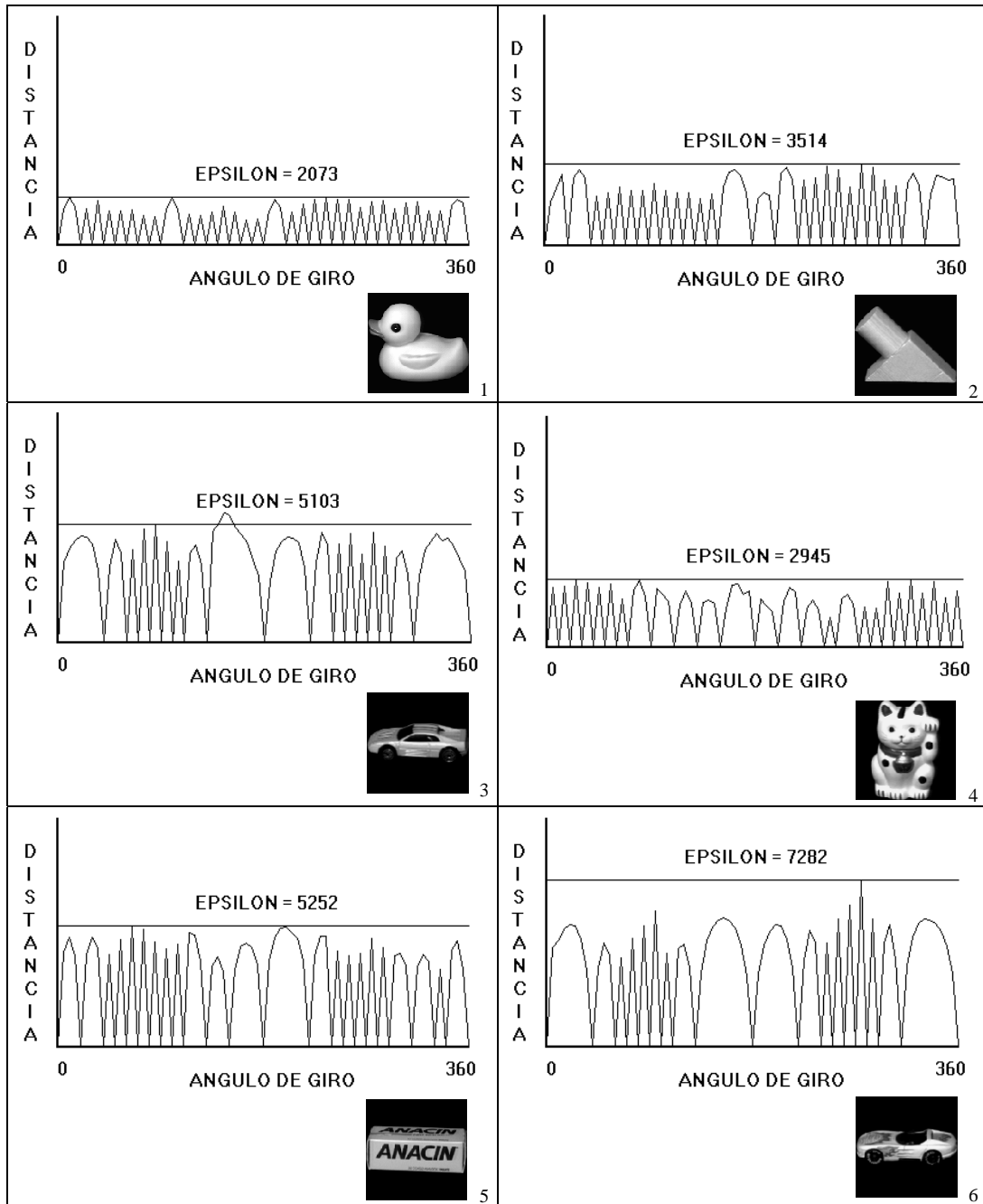


Figura 20: Graficando de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de COIL-20 (Figura 15), empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

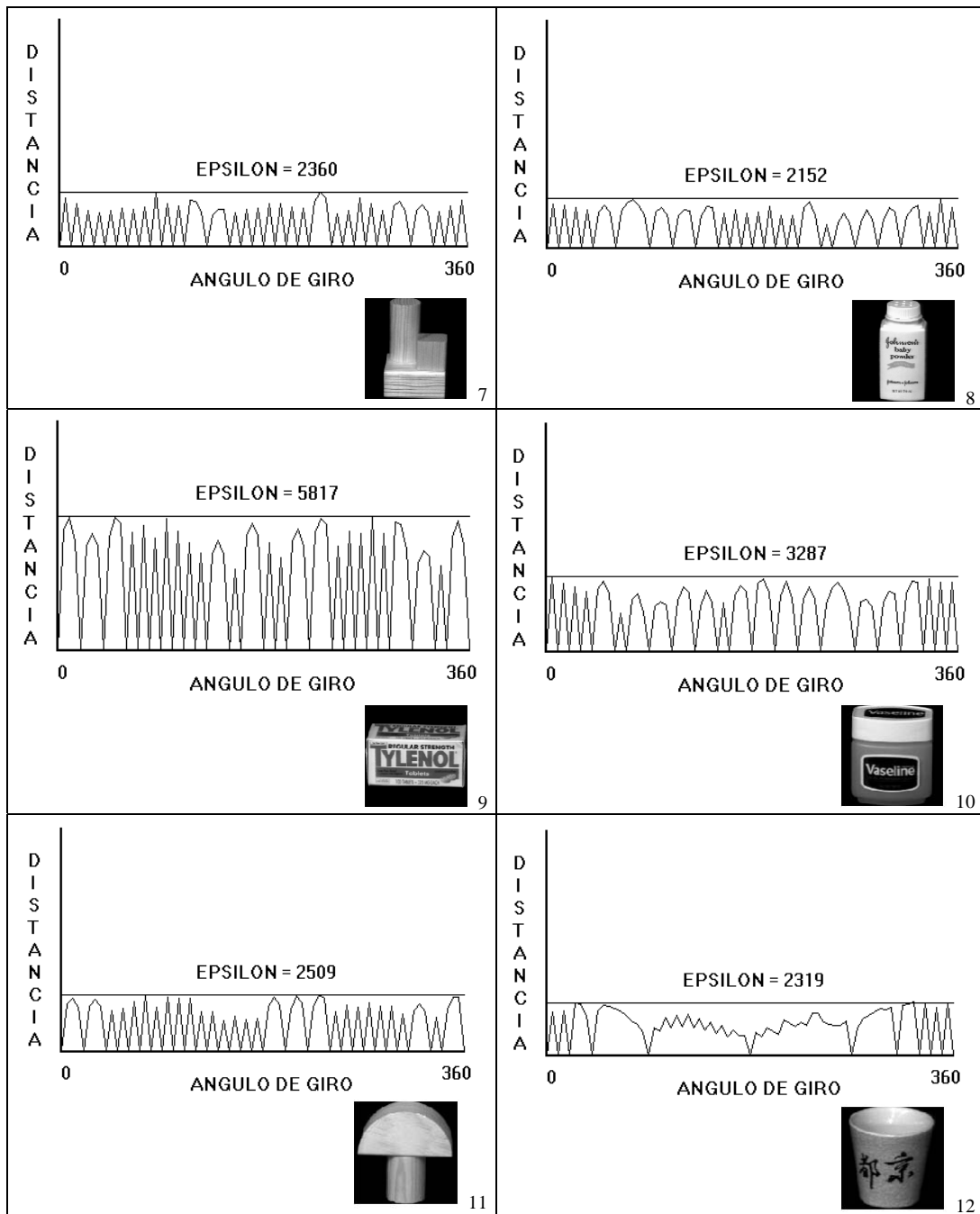


Figura 20(continuación): Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de COIL-20 (Figura 15), empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

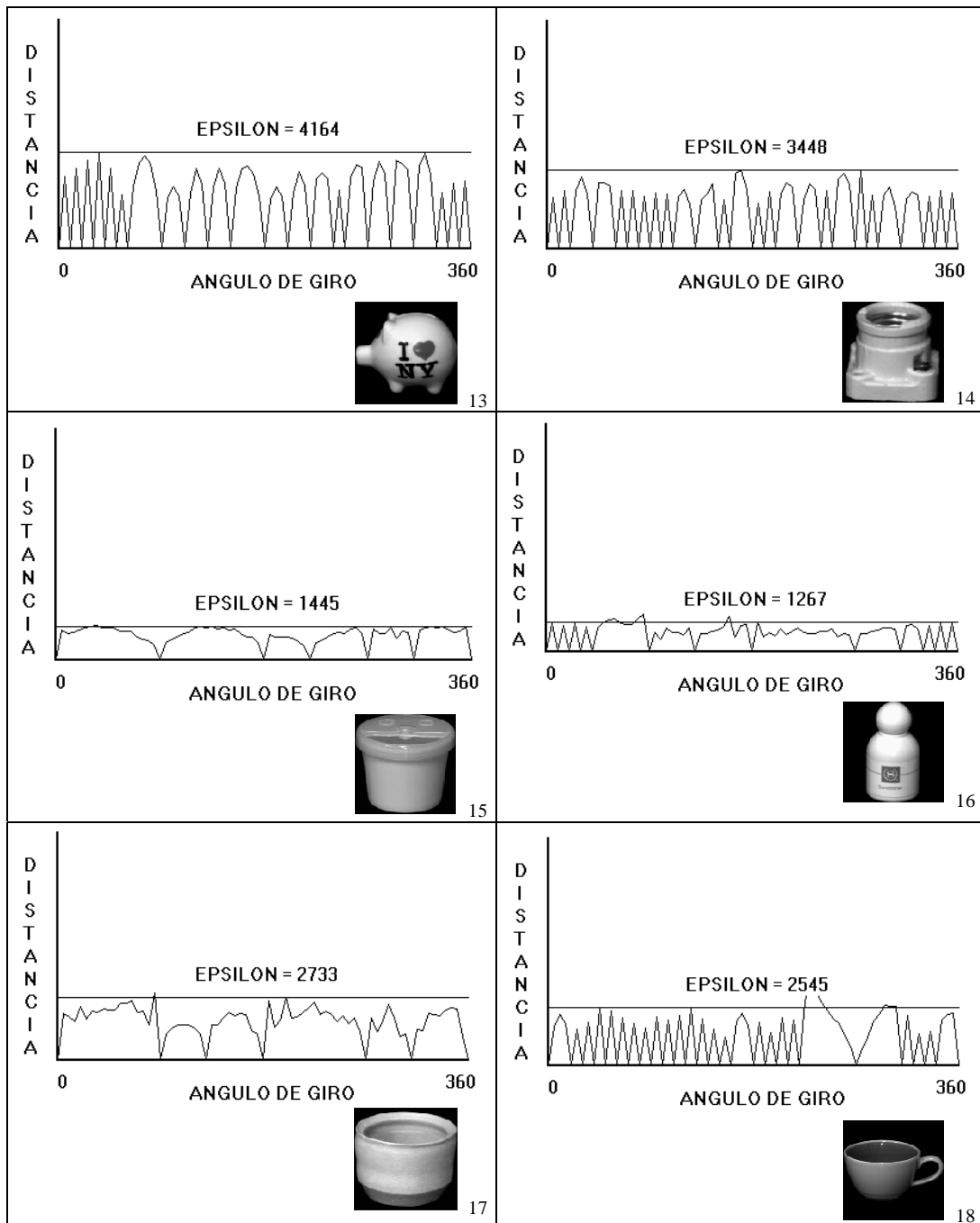


Figura 20(continuación): Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de COIL-20 (Figura 15), empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.



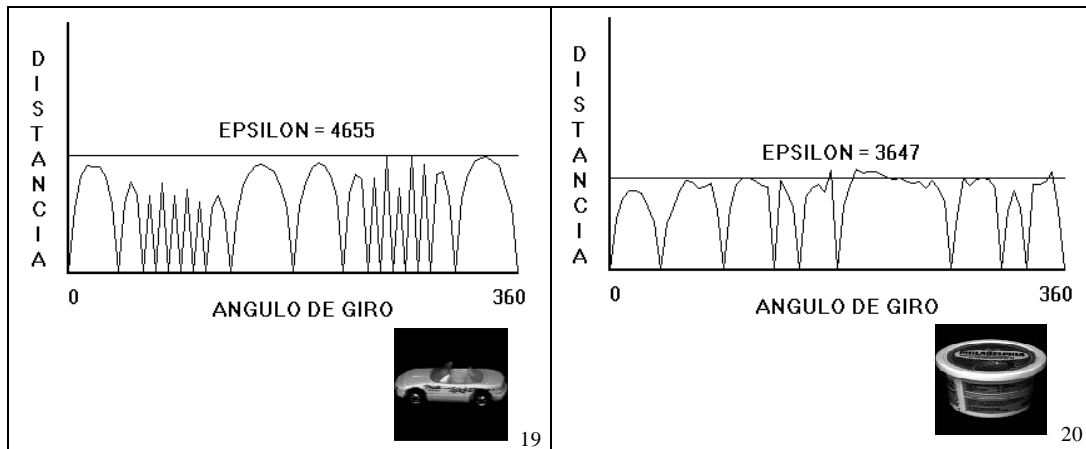


Figura 20(continuación): Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de COIL-20 (Figura 15), empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

Nótese la precisión extrema de los modelos generados por la técnica propuesta. Obsérvese que aunque en algunos modelos existen imprecisiones, éstas no son de magnitud considerable, y por tanto, los modelos son una buena aproximación de la apariencia de los objetos. En particular, obsérvese la buena calidad de los modelos obtenidos para los objetos 2, 3 y 4 de la Figura 19, así como para los objetos 12, 15, 16, 17 y 20 de la Figura 20, en donde la cantidad de imágenes empleada es pequeña.

Por supuesto, es posible lograr mejores resultados en el reconocimiento del objeto si se usan las 36 imágenes pares de cada objeto (como generalmente lo hacen los sistemas propuestos en la literatura), pero solamente en las zonas en donde la precisión del modelo es mejor que la precisión garantizada; sin embargo, en zonas donde la precisión es igual a la precisión garantizada, el reconocimiento del objeto será pobre.

Hasta donde se ha logrado investigar en la literatura, no han existido estudios previos acerca de la precisión de los modelos generados con COIL-20. Por lo tanto, trabajos previos que han usado COIL-20 para probar la efectividad de los sistemas de reconocimiento que proponen, pudieran mejorar sus resultados si tomaran en cuenta emplear modelos más precisos, como los obtenidos mediante el empleo de la presente propuesta, en lugar de los generados por un simple muestreo uniforme. Esto fue señalado en [A101].

### 6.1.6 Reducción de la cantidad de imágenes

Como se ha podido observar de los resultados experimentales presentados en las Secciones previas, se ha podido constatar que la cantidad de imágenes requeridas para modelar y reconocer objetos es dependiente del tipo de objeto a considerar, y que emplear un muestreo uniforme no garantiza precisión alguna sobre el modelo generado, y en cambio, puede exigir una gran capacidad de almacenamiento y tiempo de procesamiento, de los sistemas de reconocimiento. Con la propuesta hecha en el Capítulo 4 se ha observado en las Secciones previas que es posible subsanar los problemas anteriores, si se emplean un muestreo no-uniforme y dependiente del tipo de objeto a considerar. En la Sección previa se ha analizado la precisión de los modelos generados con esta propuesta. En esta Sección se analizará la ventaja de obtener una reducción en la cantidad de imágenes necesaria para modelar a los objetos.

	OBJETO									
	1	2	3	4	5	6	7	8	9	10
Muestreo uniforme	36	36	36	36	36	36	36	36	36	36
Muestreo no-uniforme (propuesto)	23	12	4	3	27	32	24	30	26	18
Número de imágenes no necesarias	13	24	32	33	9	4	12	6	10	18
Porcentaje de ahorro (%)	36	67	89	92	25	11	33	17	28	50
<b>Porcentaje de ahorro promedio: 45%</b>										

Tabla 6: Tabla comparativa entre un muestreo uniforme y el muestreo no-uniforme generado por la propuesta, respecto al número de imágenes requeridas para los objetos de la Figura 14.

Para modelar a los objetos de las Figuras 14 y 15 se habrían requerido 36 imágenes si se hubiera empleado un muestreo uniforme, con un espaciamiento de  $10^\circ$  entre imágenes consecutivas. Sin embargo, con un muestreo no uniforme y preservando la precisión máxima del modelo, se requirieron únicamente las cantidades listadas en la columna 4 de las Tablas 4 y 5. En las Tablas 6 y 7 se muestra en el primer renglón, la cantidad de imágenes requeridas mediante un muestreo uniforme, para modelar a los objetos de las Figuras 14 y 15, respectivamente, y en el segundo renglón las cantidades requeridas mediante el muestreo no uniforme propuesto. En el tercer renglón se calcula la diferencia entre las imágenes requeridas por ambos muestreos (renglón1 - renglón2). En el cuarto renglón se muestra el porcentaje de

ahorro en la cantidad de imágenes de la técnica propuesta, respecto a un muestreo uniforme. Finalmente, en el quinto renglón se muestra el promedio de porcentaje de ahorro para todos los objetos listados.

	OBJETO																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Muestreo uniforme	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	
Muestreo no-uniforme (propuesto)	32	27	18	26	23	18	31	25	26	22	29	11	22	26	6	13	6	26	18	9	
Número de imágenes no necesarias	4	9	18	10	13	18	5	11	10	14	7	25	14	10	30	23	30	10	18	27	
Porcentaje de ahorro (%)	11	25	50	28	36	50	14	31	28	39	19	69	39	28	83	64	83	28	50	75	
																	<b>Porcentaje de ahorro promedio:</b>				<b>43%</b>

Tabla 7: Tabla comparativa entre un muestreo uniforme y el muestreo no-uniforme generado por la propuesta, respecto al número de imágenes requeridas para los objetos de la Figura 15 (COIL-20).

De estos resultados se desprende que existe un ahorro en la cantidad de imágenes necesarias para modelar objetos, empleando la técnica propuesta, respecto a un muestreo uniforme. Este ahorro está comprendido dentro del rango de 11% hasta 92%. Evidentemente, el ahorro más sustancial se logra con objetos que presentan una baja complejidad en su apariencia y en contraste, un ahorro menos significativo es logrado en objetos con apariencia compleja. El promedio de ahorro en ambas Tablas 6 y 7 es muy similar (45% y 43%, respectivamente); esto da una pauta acerca de cuánto ahorro se puede esperar de la técnica propuesta, en términos del número de imágenes requeridas para modelar un objeto cualquiera. Como se puede notar, este ahorro esperado es muy significativo, y dado que una imagen consume espacio físico en los dispositivos de almacenamiento, el ahorro en términos de quizá Megabytes, será también una ventaja adicional de la propuesta presentada. Sin embargo, estos resultados porcentuales serán tomados con mucha reserva, hasta que se realice un análisis más profundo de ellos, en el Capítulo 8.

De estos resultados se desprende una propiedad importante, observada previamente en [AI01]: Para cualquier objeto en general, se tiene que:

- *Es posible obtener una reducción en la cantidad de imágenes requeridas para modelar un objeto cualquiera, respecto a un muestreo uniforme, preservando la precisión máxima, ó bien,*

- Usando la misma cantidad de imágenes que para un muestreo uniforme, es posible mejorar la precisión máxima del modelo.

## 6.2 Múltiples parámetros

En esta sección se exponen los principales resultados experimentales obtenidos con la técnica generalizada (algoritmo 2), expuesta en el Capítulo 5. Para obtener estos resultados se emplearon 2 parámetros para determinar la apariencia de los objetos; el primer parámetro quedó determinado por el ángulo de giro de la tornamesa, del sistema de hardware para la adquisición de imágenes. El segundo quedó determinado por el ángulo de elevación/descenso de la cámara, sobre el plano de la tornamesa. Tal elevación/descenso fue realizado describiendo una semi-circunferencia centrada en el centro de la tornamesa, de tal manera que la distancia del sensor al centro de la tornamesa siempre es constante. Para los experimentos realizados, esta distancia fue de aproximadamente 30 cm., y se limitó el análisis de los objetos al rango de  $0^\circ - 20^\circ$  de elevación del sensor. Las imágenes que adquirió el sistema posicionador fueron segmentadas del fondo, y normalizadas en escala para ser de  $128 \times 128$  píxeles. Un ejemplo de la forma de adquirir imágenes con 2 parámetros, es ilustrado en la Figura 21, empleando un muestreo uniforme.



Figura 21: Un ejemplo de la forma de adquirir imágenes para un objeto, para construir su modelo bajo 2 parámetros, empleando un muestreo uniforme con  $45^\circ$  de incremento en el ángulo de giro de la tornamesa y  $5^\circ$  de incremento en el ángulo de elevación de la cámara, respecto al plano de la tornamesa, entre imágenes consecutivas.

### 6.2.1 Objetos de estudio

Para realizar los experimentos, se emplearon los objetos mostrados en la Figura 14 (Sección 6.1.1).

### 6.2.2 Determinación del número de imágenes requeridas

Como una primera etapa, se procedió a aplicar el algoritmo propuesto sobre los objetos mostrados en las Figuras 14, con diversos valores para la precisión deseada  $\epsilon$ . Los resultados del experimento son mostrados en la Tabla 8. En ellas, las celdas etiquetadas con + indican que para alcanzar la precisión requerida  $\epsilon$ , se necesitaba que el incremento en el ángulo de giro de la tornamesa o en el ángulo de elevación/descenso de la cámara, entre imágenes sucesivas fuera menor a  $5^\circ$ . Sin embargo, para este experimento, se limitó al sistema de adquisición de imágenes a no superar esta precisión (en la literatura típicamente se reporta una precisión de entre  $12^\circ$  a  $10^\circ$  como máximo; ver, por ejemplo, [Na96a]). Para la implementación del algoritmo generalizado, se empleó una división heterogénea de zonas (Ver Sección 5.5). Aún cuando en la Tabla 8 no se muestra, el algoritmo determinó, adicionalmente al número de imágenes necesarias, cuales son las imágenes estrictamente necesarias para cada  $\epsilon$ , de la misma manera que se mostró en la Sección 6.1.2, para el caso de un parámetro. Sin embargo, por razones de espacio no se listarán aquí.

		OBJETO									
		1	2	3	4	5	6	7	8	9	10
	$\epsilon = 2000$	+	+	+	+	+	+	+	+	+	+
	$\epsilon = 2500$	+	+	12	11	+	+	+	+	+	+
<b>P</b>	$\epsilon = 3000$	+	+	9	11	+	+	95	+	+	+
<b>R</b>	$\epsilon = 3500$	+	18	4	9	+	94	72	84	+	+
<b>E</b>	$\epsilon = 4000$	+	11	4	4	+	67	52	69	+	+
<b>C</b>	$\epsilon = 4500$	+	11	4	4	+	60	47	52	+	+
<b>I</b>	$\epsilon = 5000$	+	9	4	4	70	49	43	47	+	+
<b>S</b>	$\epsilon = 5500$	+	4	4	4	63	42	37	41	59	11
<b>I</b>	$\epsilon = 6000$	+	4	4	4	58	34	29	32	53	9
<b>O</b>	$\epsilon = 6500$	+	4	4	4	44	30	29	32	36	4
<b>N</b>	$\epsilon = 7000$	58	4	4	4	38	27	26	29	30	4
	$\epsilon = 7500$	46	4	4	4	33	24	22	23	24	4
	$\epsilon = 8000$	41	4	4	4	25	21	18	20	23	4

Tabla 8: Número de imágenes requeridas para capturar la apariencia de los objetos en la Figura 14, dependiendo de distintos valores de  $\epsilon$ , y empleando 2 parámetros (rotación de la tornamesa y elevación de la cámara). Las celdas marcadas con + indican que para alcanzar la precisión fijada por  $\epsilon$ , el ángulo de rotación de la tornamesa ó el ángulo de elevación de la cámara entre imágenes sucesivas, debe ser inferior a  $5^\circ$ .

Analizando los resultados mostrados en la Tabla, se puede corroborar que, como en el caso de un único parámetro, la adquisición de imágenes depende de la precisión deseada  $\epsilon$  que el investigador desee, y de la clase de objeto a ser considerado. De esta forma, para lograr alcanzar una determinada precisión en el modelo, se requiere una cantidad distinta de imágenes para objetos diferentes. Observando los resultados obtenidos, se constata la siguiente propiedad importante (análoga al caso de un único parámetro): a mayor precisión exigida para el modelo ( $\epsilon$  pequeño), mayor cantidad de imágenes. Esta propiedad contrasta fuertemente con los resultados encontrados en la literatura, que suponen que una cantidad fija de imágenes, es suficiente para capturar la apariencia de cualquier objeto. Otro hecho interesante que se puede observar de estos resultados, radica en que el rango de valores permitidos para la precisión  $\epsilon$  (esto es, el rango de valores que no rebasan la precisión del hardware), es propia de cada objeto en particular. La determinación de tal rango de manera automática para cada objeto particular, será abordada en el Capítulo 8.

### **6.2.3 Distribución de las imágenes estrictamente necesarias**

Aún cuando los resultados mostrados en la Tabla 8 son significativos, no permiten realizar un análisis completo de los resultados obtenidos. Por esta razón, en la Figura 22 se ha graficado la distribución de las imágenes determinadas por el sistema, empleando la precisión máxima (como fue explicado en la Sección 6.1.3), para el caso de 2 parámetros. En esta Figura, el Eje X representa el ángulo de giro de la tornamesa, y el eje Y el ángulo de elevación de la cámara. Cada intersección de líneas (denotada por un pequeño círculo) señala la posición de una imagen adquirida para construir el modelo.

De estos resultados se puede observar cómo el algoritmo generalizado propuesto es capaz de localizar las zonas donde la apariencia varía poco y en consecuencia adquirir pocas imágenes; de una manera similar, es capaz de identificar las zonas con grandes variaciones en la apariencia y, en contraste, adquirir una gran cantidad de imágenes. En particular nótese la poca cantidad de imágenes adquiridas para los objetos 2, 3, 4 y 10 de la Figura 22, cuya apariencia es poco compleja. Otro resultado importante que se puede observar de estas distribuciones, se refiere al efecto de la normalización de imágenes sobre los modelos de los objetos. Este efecto ya fue observado y expuesto para el caso de un parámetro (Ver Sección 6.1.4) y aquí se vuelve a observar. Las distribuciones para los objetos 1, 5, 6, 7, 8 y 9 de la Figura 22, denotan la alta concentración de las imágenes adquiridas en las zonas donde se requiere un escalado considerable de las imágenes.

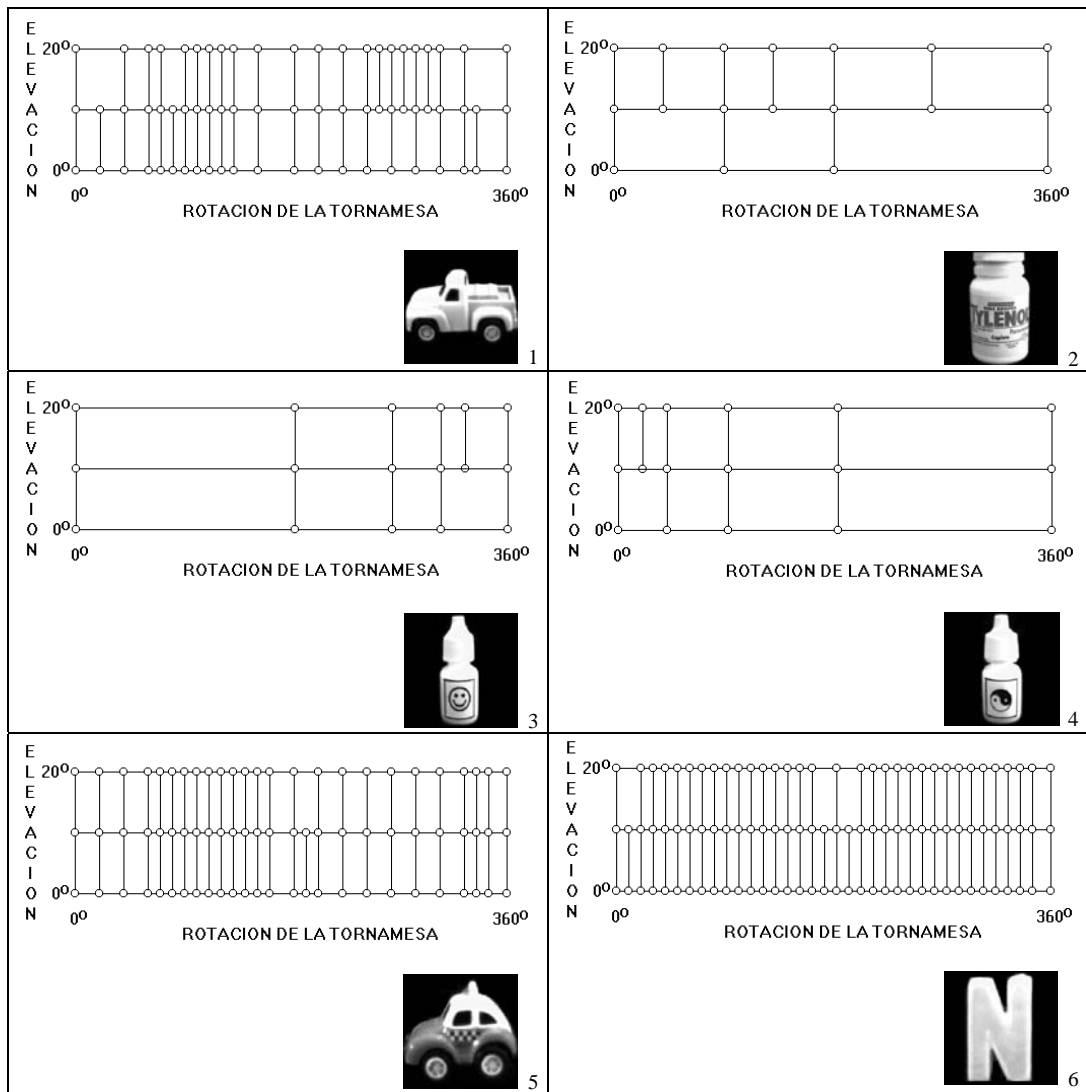


Figura 22: Distribución de las imágenes necesarias para modelar la apariencia de los objetos de la Figura 14, bajo 2 parámetros: rotación de la tornamesa (eje X) y elevación de la cámara (eje Y).

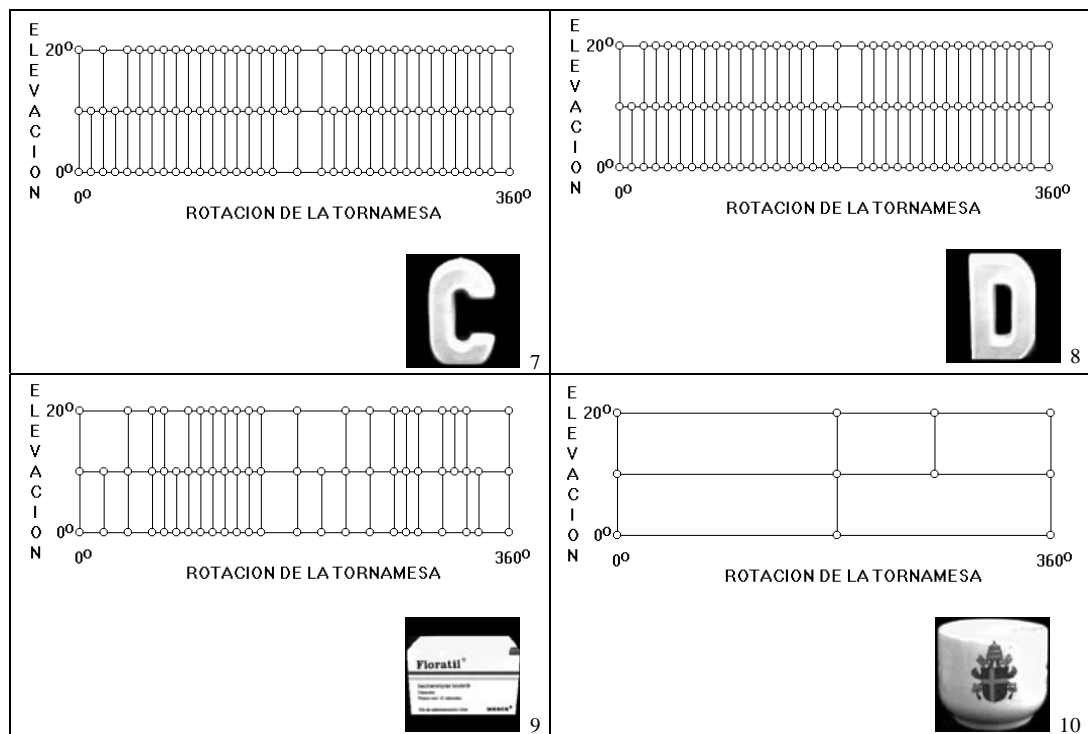


Figura 22(continuación): Distribución de las imágenes necesarias para modelar la apariencia de los objetos de la Figura 14, bajo 2 parámetros: rotación de la tornamesa (eje X) y elevación de la cámara (eje Y).

#### 6.2.4 Precisión de los modelos

Se probó la calidad de los modelos generados mediante la técnica propuesta generalizada, constatándose los mismos resultados que para el caso de un solo parámetro: la calidad de los modelos generados es buena, aún cuando existen ligeras desviaciones en algunas zonas del modelo. Sin embargo, como la magnitud de las desviaciones es pequeña, se considera al modelo como de buena calidad, para propósitos de reconocimiento de objetos. Para el caso de 2 parámetros no fueron graficados los resultados, debido a que la visualización es más complicada que con un solo parámetro, y los resultados solamente fueron analizados numéricamente.

#### 6.2.5 Reducción de la cantidad de imágenes

Evidentemente, si el ahorro en la cantidad de imágenes obtenido para el caso de un solo parámetro fue significativo, para el caso de 2 parámetros se espera un ahorro considerable. Esto se debe a que la cantidad de imágenes involucrada en este caso es aún mayor (Ver Sección 3.1).



	OBJETO									
	1	2	3	4	5	6	7	8	9	10
Muestreo uniforme	108	108	108	108	108	108	108	108	108	108
<b>Muestreo no-uniforme (propuesto)</b>	69	18	17	17	77	105	101	103	70	11
<b>Número de imágenes no necesarias</b>	39	90	91	91	31	3	7	5	38	97
<b>Porcentaje de ahorro (%)</b>	36	83	84	84	29	3	6	5	35	90
<b>Porcentaje de ahorro promedio:</b>										<b>46%</b>

Tabla 9: Tabla comparativa entre un muestreo uniforme y el muestreo no-uniforme generado por la propuesta, respecto al número de imágenes requeridas para los objetos de la Figura 14, empleando 2 parámetros.

Para modelar a los objetos de la Figura 14 se habrían requerido 108 imágenes si se hubiera empleado un muestreo uniforme, con un espaciamiento de  $10^\circ$  entre imágenes consecutivas ( $36 \times 3$ ), para ambos parámetros. En la Tabla 9 se muestra en el primer renglón, la cantidad de imágenes requeridas mediante un muestreo uniforme, para modelar a los objetos de la Figura 14, y en el segundo renglón se listan las cantidades requeridas por el muestreo no uniforme propuesto, empleando la precisión máxima, calculada ésta como se indicó en la Sección 6.1.3, pero ahora sobre ambos parámetros. En el tercer renglón se calcula la diferencia entre las imágenes requeridas por ambos muestreos (renglón1 - renglón2). En el cuarto renglón se muestra el porcentaje de ahorro en la cantidad de imágenes de la técnica propuesta, respecto a un muestreo uniforme. Finalmente, en el quinto renglón se muestra el promedio de porcentaje de ahorro para todos los objetos listados.

De estos resultados se desprende que existe un ahorro en la cantidad de imágenes necesarias para modelar objetos, empleando la técnica propuesta, respecto a un muestreo uniforme. Este ahorro está comprendido dentro del rango de 3% hasta 97%. Evidentemente, el ahorro más sustancial se logra con objetos que presentan una baja complejidad en su apariencia y en contraste, un ahorro menos significativo es logrado en objetos con apariencia compleja. El promedio de ahorro es muy similar al encontrado con un solo parámetro (46%); esto da una pauta acerca de cuánto ahorro se puede esperar de la técnica propuesta, en términos del número de imágenes requeridas para modelar un objeto cualquiera, bajo múltiples parámetros. Como se puede notar, este ahorro esperado es muy significativo, y dado que una imagen consume espacio físico en los dispositivos de almacenamiento, el ahorro en términos de quizá Megabytes, será también una ventaja adicional de la propuesta presentada, en

especial si se considera más de un parámetro. Sin embargo y al igual que en el caso de un parámetro, estos resultados porcentuales serán tomados con mucha reserva, hasta que se realice un análisis más profundo de ellos, en el Capítulo 8.

# CAPÍTULO 7

## *APLICACIONES*

**Resumen:** En este Capítulo se expone una posible aplicación industrial de las ideas planteadas en la presente tesis. Tal aplicación tiene fundamento en un problema real que forma parte de un proyecto que desarrolla el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) para la compañía armadora de automóviles Volkswagen de México. Si bien las ideas planteadas en esta tesis no pueden ser empleadas directamente para resolver este problema, debido a que se requiere resolver algunos otros problemas colaterales propios de una aplicación real, si se pretende dar en este Capítulo un esbozo de las aplicaciones potenciales de la presente propuesta de tesis.

---

Diversos investigadores opinan que los enfoques basados en apariencia, podrían ser un valioso complemento de las actuales propuestas basadas en otros enfoques [P&H96a], [Na96a], [Mu96], [Tu91]. Es por ello que, en general, se espera que la presente propuesta pueda ser utilizada en todas aquellas aplicaciones donde se empleen enfoques basados en apariencia, pues en todas ellas sería de gran valía obtener mejores índices de reconocimiento, mediante el empleo de modelos más precisos de los objetos, además de obtener una reducción importante en la cantidad de imágenes necesarias y en el tiempo de cómputo requerido.

Algunas de las aplicaciones concretas pueden ser aquellas listadas en [Na96a], y que incluyen: reconocimiento de objetos en tiempo real, estimación de parámetros de posición del objeto respecto al sensor, posicionamiento de robots sobre partes de objetos como tabletas de circuitos para el ensamble de circuitos integrados y seguimiento de objetos en escenas dinámicas.

Finalmente, la aplicación concreta que actualmente se está implementando, se realiza dentro del marco de un proyecto de aplicación que desarrolla el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), para la empresa armadora de automóviles Volkswagen. La dirección de este proyecto se encuentra a cargo del Dr. Leopoldo Altamirano R.

### **7.1 Descripción**

Brevemente, este proyecto consiste en lo siguiente: el chasis de cualquier automóvil ensamblado en la planta, debe contener una cierta cantidad de pernos industriales, que son soldados al chasis mediante robots. Los tipos de pernos son de

distintas clases, algunas parecidas entre sí. Por la acción de diversos factores (mecánicos, humanos, etc.), los robots no soldan los pernos correctamente en la posición que debieran, ni colocan el tipo de perno correcto que debe ir en ese punto. Estos errores son detectados actualmente mediante inspección ocular, y por tanto, el proceso de revisión es lento y sujeto a múltiples errores humanos. Bajo el marco de este proyecto se pretende automatizar el proceso de inspección de las carrocerías de los automóviles.

## **7.2 Trabajo previo**

Actualmente, se cuenta con un sistema que es capaz de determinar mediante procesamiento geométrico, si un perno se encuentra soldado ó no, en una posición específica en la carrocería del vehículo. Sin embargo, no es capaz de determinar si la posición está dentro de la tolerancia permitida por las normas de la planta. Además, el problema de determinar si el tipo de perno soldado es el correcto ó no, aún no ha sido resuelto satisfactoriamente, debido a que la geometría de los pernos es muy similar. Debido a que mediante la apariencia se pueden capturar una gran cantidad de detalles (tanto geométricos como de textura, color, etc.), se espera que este problema pueda ser solucionado de una forma más óptima que la que actualmente se tiene, mediante el empleo de modelos basados en apariencia.

## **7.3 Justificación del empleo de la propuesta**

La presente propuesta es aplicable al problema descrito, debido a que no se sabe cuál es la precisión que se debiera emplear al momento de construir los modelos de los pernos. Es en este punto en donde se pretende emplear la técnica propuesta en este documento, para lograr una mejor diferenciación entre los distintos tipos de pernos, aún sobre los que son muy semejantes entre sí. Es muy importante considerar la precisión de los modelos, debido a que como son muy semejantes, el sistema podría erróneamente señalar que el tipo de perno soldado es el correcto, o bien, que no está soldado en la posición correcta, dentro de los límites tolerables por las normas de la planta. Esto repercutirá en severos retrasos en los siguientes pasos del proceso de ensamblado de los vehículos, y como consecuencia, implicará un descenso en las normas de calidad de la planta. El reconocimiento preciso es también importante, porque uno de los objetivos que deberá cumplir el sistema implementado será incrementar la producción de vehículos, y si el sistema incurre frecuentemente en errores, este objetivo no podrá ser alcanzado. Este quizá es el punto más importante para la planta ensambladora.

## **7.4 Resultados experimentales**

Aún cuando los resultados aquí reportados son solamente preliminares, debido a que para obtener una aplicación industrial hace falta resolver algunos problemas

adicionales, si pretenden dar una idea de los resultados potenciales que pueden ser obtenidos mediante el empleo de modelos basados en apariencia, junto con el control de la precisión en ellos mediante la propuesta realizada, dentro de una aplicación industrial.

#### 7.4.1 Objetos de estudio

Aún cuando en una carrocería se pueden tener varios tipos de pernos, se realizaron los experimentos aquí citados, únicamente sobre los 3 pernos mostrados en la Figura 23, por cuestiones de tiempo. Todas las imágenes de estos pernos empleadas en este estudio, fueron normalizadas en escala, para ser de  $128 \times 128$  píxeles.

#### 7.4.2 Parámetros empleados

Para definir los modelos a estudiar de los pernos mostrados en la Figura 23, se empleó únicamente un parámetro, definido éste mediante el ángulo de giro de la tornamesa. Dado que los pernos a estudiar fueron situados con uno de sus extremos aproximadamente en el centro de la tornamesa, algunas de las diferentes posiciones en que puede aparecer el perno en una carrocería son esbozadas en la Figura 24. La elección del parámetro de estudio fue realizada en esta forma, para poder estudiar la eficiencia del sistema tanto para determinar tipo de perno, como posición de éste en la imagen. La posición permitirá al sistema decidir si el perno está soldado con una desviación tolerable, por las normas de la empresa. Evidentemente, para una aplicación industrial, se debiera considerar al menos 2 parámetros (posición, inclinación, etc.), y emplearse la propuesta generalizada descrita en el Capítulo 5, sin embargo, para propósitos experimentales únicamente se empleó el parámetro descrito.

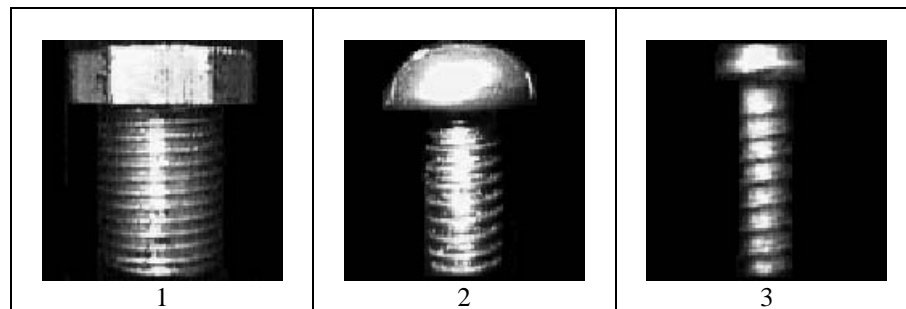


Figura 23: Pernos industriales estudiados.

#### 7.4.3 Adquisición adaptativa de imágenes

Se procedió a aplicar el sistema de software obtenido, sobre los pernos industriales par poder construir sus modelos, limitando al sistema a no adquirir

imágenes consecutivas con un ángulo de giro de menos de  $5^\circ$ , y empleando la precisión máxima para cada perno, como fue explicado en el Capítulo 6. Las distribuciones obtenidas mediante este procedimiento para cada perno, son mostradas en la Figura 25. Puede observarse que, a diferencia de lo supuesto por un muestreo uniforme, las distribuciones y la cantidad de imágenes requeridas, son distintas para cada perno, y justamente la técnica propuesta identifica este hecho.

	POSICIÓN							
	$0^\circ$	$45^\circ$	$90^\circ$	$135^\circ$	$180^\circ$	$225^\circ$	$270^\circ$	$315^\circ$
Perno 1								
Perno 2								
Perno 3								

Figura 24: Algunas posiciones en que puede aparecer un perno soldado a una carrocería.

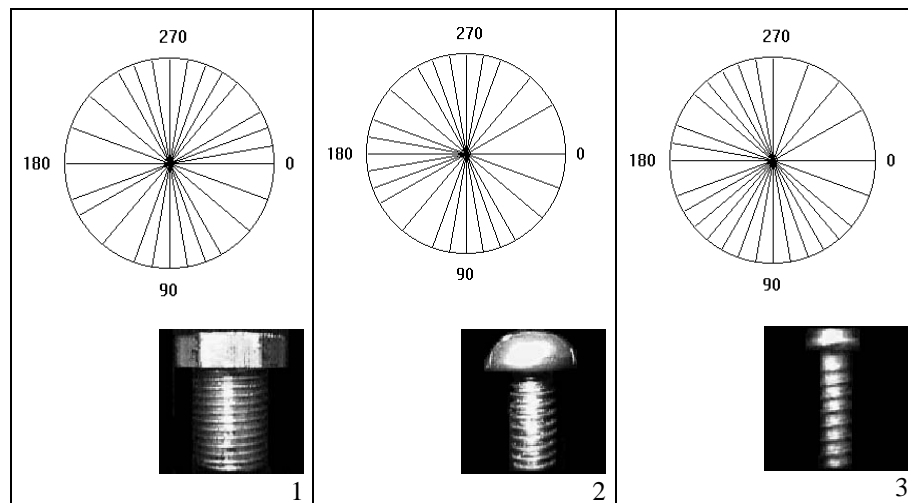


Figura 25: Distribuciones de las imágenes requeridas para modelar los pernos estudiados.

#### 7.4.4 Precisión de los modelos

Se probó la calidad de los modelos obtenidos, empleando únicamente las imágenes estrictamente necesarias, determinadas por el sistema desarrollado. Esto se realizó de la misma forma en que fue descrito en el Capítulo previo (Ver Sección 6.1.5). Los resultados de esta prueba son mostrados en la Figura 26. De estos resultados se puede observar que los modelos obtenidos, cumplen con las restricciones de precisión exigidos, en todas las zonas del modelo. Por tanto, los objetos son bien aproximados por el modelo, en todas las posiciones definidas por el parámetro de estudio.

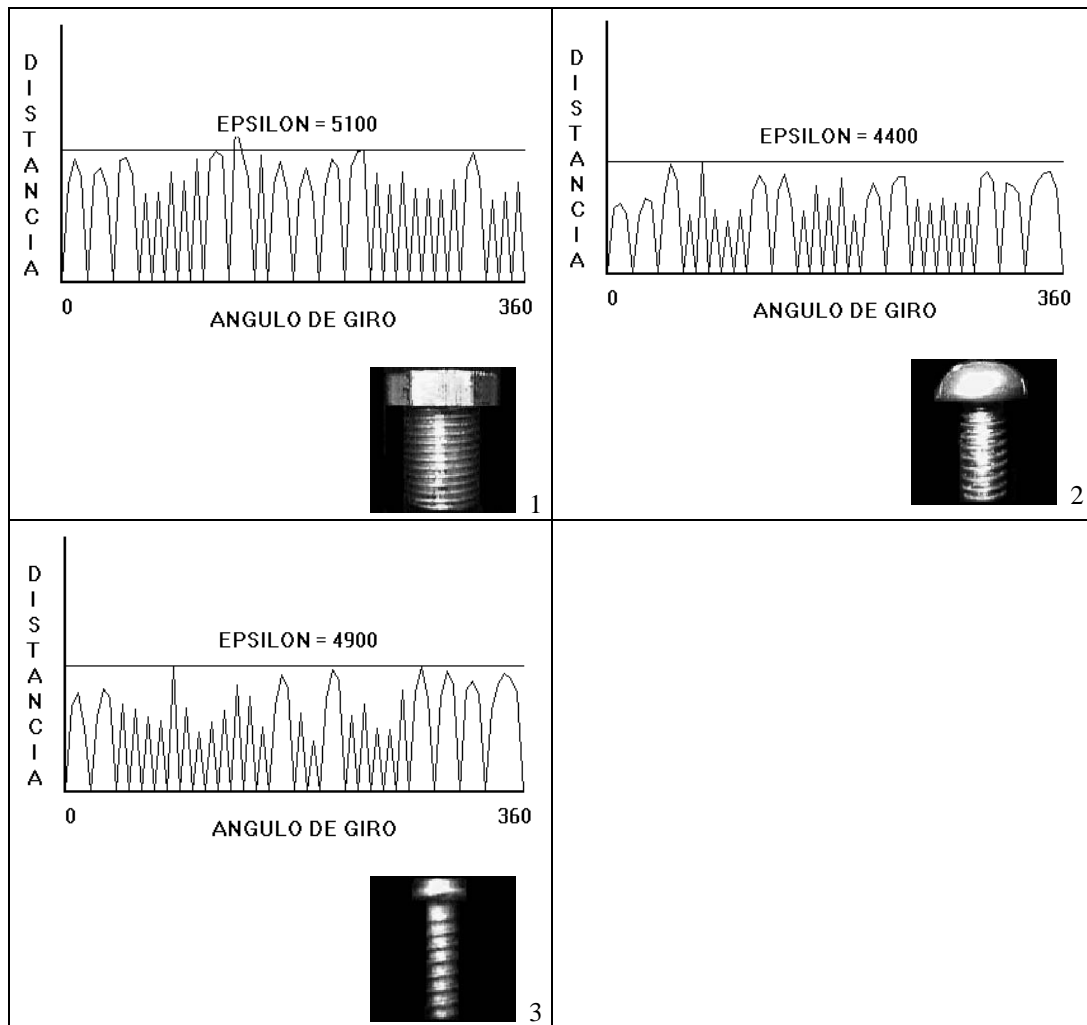


Figura 26: Graficando de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los pernos de la Figura 23, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

### 7.4.5 Reducción de la cantidad de imágenes

Empleando la propuesta realizada sobre los pernos de estudio, se obtuvieron reducciones del 28% (26 imágenes), 31% (25 imágenes) y 25% (27 imágenes), para los pernos 1, 2 y 3, respectivamente, de la Figura 23, respecto a un muestreo uniforme de  $10^\circ$  entre imágenes consecutivas. Evidentemente, estas reducciones son significativas.

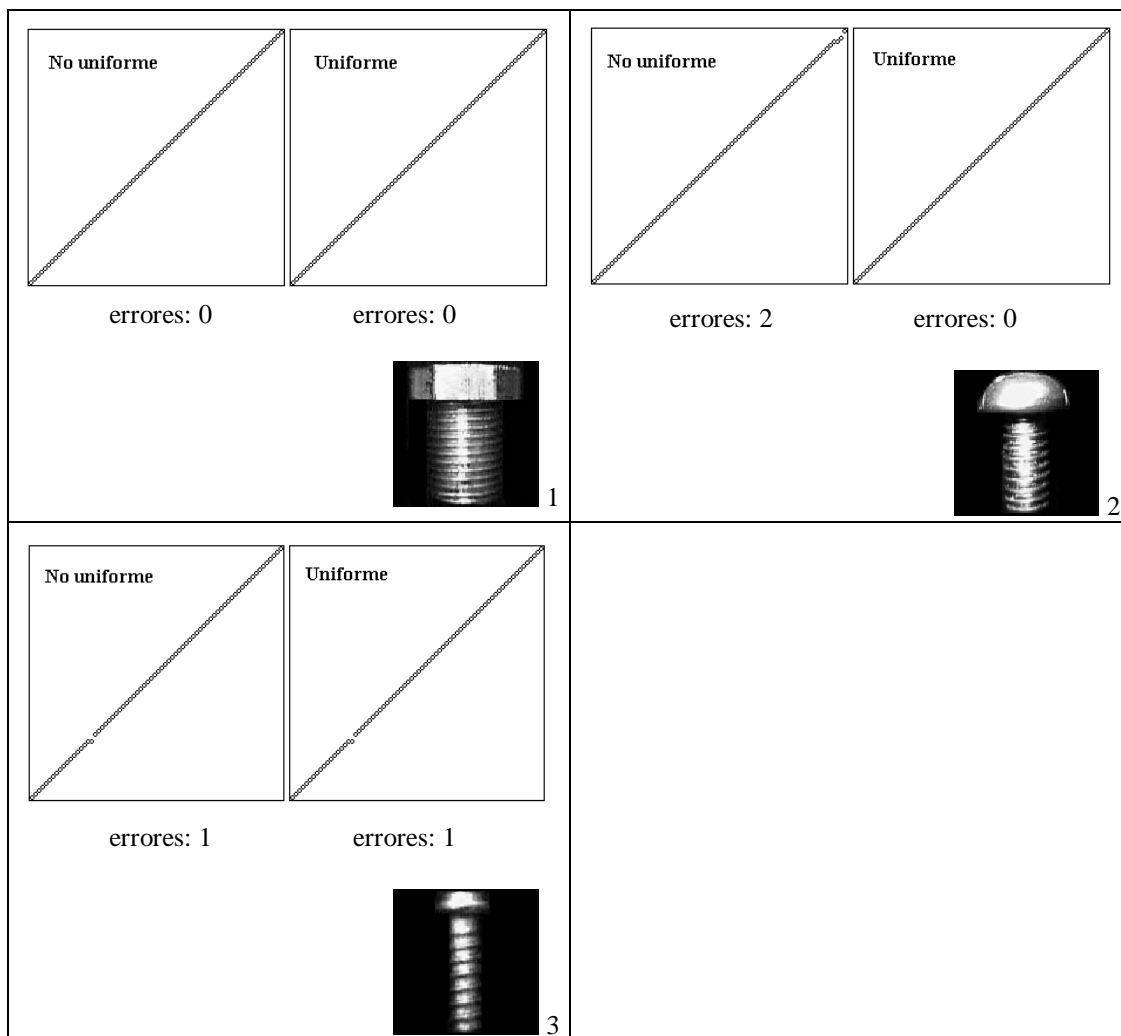


Figura 27: Graficado de los resultados obtenidos al estimar la posición de los objetos mostrados en la Figura 23 en 72 imágenes por objeto. La estimación de posición será precisa si la posición real es igual a la estimada por el sistema, por tanto, una línea recta en una gráfica denota una estimación precisa para todas las imágenes.

Eje X: posición real del objeto. Eje Y: posición estimada por el sistema.



#### 7.4.6 Reconocimiento

Los modelos generados fueron empleados para reconocer a los pernos en posiciones desconocidas. Esto se realizó en 2 etapas: en la primera, se adquirieron 72 imágenes de cada perno, uniformemente espaciadas a 5° grados entre imágenes consecutivas. Estas imágenes fueron empleadas para probar la eficiencia en el reconocimiento del sistema. Los resultados para cada uno de los pernos analizados son mostrados en la Figura 27. El porcentaje de reconocimiento, en base al número de aciertos en la estimación del parámetro fue de aproximadamente 98% para esta etapa.

En la segunda etapa, se adquirieron 20 imágenes de cada perno, en posiciones aleatorias y nunca aprendidas por el sistema. La posición del perno fue estimada y almacenada también, junto con la imagen correspondiente, con una precisión de aproximadamente 2.5°. Enseguida le fueron presentadas al sistema para que determinara únicamente la identidad del objeto mostrado. El sistema fue capaz de identificar al perno correcto en todas las imágenes que le fueron presentadas (porcentaje de reconocimiento = 100%). Así mismo, estas imágenes fueron empleadas para determinar la posición en que se encontraba el perno. Para hacer esto, el modelo fue discretizado para obtener muestras de él, cada 2.5° de incremento del parámetro. Enseguida se probaron nuevamente las 60 imágenes adquiridas, para determinar la posición del perno en la imagen. En este caso, en el 97% de los casos presentados al sistema, la estimación de la posición fue correcta.

Un aspecto importante a señalar, es que aunque los resultados experimentales obtenidos son excelentes, no deben de considerarse como el índice de reconocimiento del sistema. Esto se debe a que las condiciones en que se realizaron las pruebas, fueron las idóneas para el sistema. Evidentemente, el reconocimiento en una aplicación industrial no presenta estas condiciones idóneas, pues aspectos como el fondo no controlado (que para las pruebas fue controlado) y suciedad en los pernos a reconocer, son factores que influyen seriamente en el porcentaje de eficiencia del sistema. En particular, el problema del fondo no controlado, hace que los sistemas de reconocimiento dependan fuertemente de poderosos algoritmos de segmentación, como un paso previo a la realización del reconocimiento. Esto está cambiando paulatinamente con técnicas como la de Eigen-Window de Ohba e Ikeuchi [Oh96] y más recientemente, la propuesta por Leonardis y Bischof [Le00], en donde se realiza el reconocimiento sin segmentación previa. Sin embargo, la consideración de estos factores va más allá de los objetivos de la presente tesis, debido a que ellos representan un área activa de estudio en sí mismos. A pesar de esto, se está actualmente trabajando en la incorporación de la propuesta de Leonardis y Bischof [Le00], al sistema desarrollado en esta tesis.

# CAPÍTULO 8

## CONSIDERACIONES FINALES

**Resumen:** Como parte final de este trabajo de tesis, se expondrán algunos aspectos muy importantes, relacionados con la propuesta realizada y que contribuirán a mejorarla.

---

### 8.1 Detección y corrección de errores en el modelo

Debido a que la técnica propuesta en los Capítulos 4 y 5 garantiza que el modelo que se obtiene al emplearla, cumple con los requerimientos de precisión  $\epsilon$  en el punto medio de cada subintervalo que analizó, puede incurrir en errores de precisión en zonas distintas al punto medio de estos subintervalos. Esto fue señalado en la Sección 6.1.5. Estos errores consisten en que el modelo se aleja de la apariencia del objeto, más allá de la precisión  $\epsilon$  exigida, es decir, la distancia entre el modelo y la apariencia del objeto es mayor que el  $\epsilon$  permitido, en algunas zonas (*zonas imprecisas*). Ejemplos de esta clase de imprecisiones, fueron observados en los modelos generados con la propuesta presentada, en los objetos 2, 3, 4, 6, 7, 8 y 10 de la Figura 19, y 3, 12, 15, 16, 17, 18 y 20 de la Figura 20. Si bien el grado de desviación de estas imprecisiones no es considerable y por tanto, no implicarán serios errores en el reconocimiento del objeto, es deseable no tenerlas dentro del modelo de un objeto. Por tanto, garantizar el cumplimiento de la precisión  $\epsilon$  en todas las zonas del modelo de cualquier objeto, será el objetivo de las ideas expuestas en la presente Sección.

Los errores en que incurre la propuesta presentada, se deben a que, para cada subintervalo que la técnica analiza, únicamente el punto medio se sujeta a la restricción de precisión. Entonces, cualquier otro punto dentro del subintervalo, - diferente de los puntos de los extremos, los cuales son puntos sobre la apariencia del objeto, y por tanto su distancia a la apariencia del objeto es 0, y por supuesto, diferente también del punto medio, el cual cumple con la restricción de precisión  $\epsilon$  -, puede alejarse de la apariencia del objeto más allá de la distancia permitida  $\epsilon$ . Puede observarse fácilmente, que estas imprecisiones pueden ocurrir únicamente en 3 casos distintos, los cuales son ilustrados en la Figura 28. En el primer caso, la imprecisión en el modelo se encuentra localizada en la zona inferior (mitad inferior) del subintervalo analizado por la técnica propuesta. En el segundo caso, la imprecisión en el modelo se encuentra localizada en la zona superior (mitad superior) del subintervalo. Finalmente, el tercer caso es una

combinación de los 2 casos anteriores, pues ambas zonas, inferior y superior, son imprecisas.

Para corregir estas imprecisiones (en los 3 casos), se propone el empleo del proceso predictor-corrector, ilustrado en la Figura 29. Este proceso está formado por dos partes: la primera de ellas es la adquisición adaptativa de imágenes descrita en los Capítulos 4 y 5; la segunda es un algoritmo que verifica que la precisión  $\epsilon$  está presente en todas las partes del modelo.

El funcionamiento del proceso es el siguiente: Se le proporciona un intervalo inicial de análisis del objeto, al algoritmo 1 (Capítulo 4), para obtener como salida un modelo estimado de la apariencia del objeto, bajo una precisión  $\epsilon$ , que será preciso en los puntos medios de los intervalos determinados por las posiciones de las imágenes, estimadas como indispensables para la construcción del modelo. Enseguida, este modelo estimado es discretizado y los puntos obtenidos (imágenes), son comparados con las correspondientes imágenes reales del objeto, obtenidas por el sistema de hardware para la adquisición de imágenes, para determinar si el modelo cumple con los requerimientos de precisión en todas las zonas. Esto es realizado por un algoritmo simple de determinación de distancias (Ver Figura 29). Si existe al menos una zona imprecisa, se determina el subintervalo del modelo que la contiene, y se almacena la imagen en el punto medio, como indispensable para la construcción del modelo, y se divide el intervalo en dos subintervalos (Ver Figura 29). Estos dos subintervalos son proporcionados, por separado, como entrada al algoritmo adaptativo propuesto para que realice un análisis sobre ellos y determine una mejor estimación del modelo en la zona o zonas imprecisas. Si alguna de las 2 zonas no contiene imprecisiones (casos 1 y 2), el algoritmo adaptativo determinará que no se requieren más imágenes para esa zona y únicamente analizará la zona imprecisa. En cambio, si ambas zonas son imprecisas (caso 3), las analizará por separado para mejorar la estimación de ambas. Después de este análisis, el modelo obtenido será más preciso que el previamente estimado únicamente por el algoritmo adaptativo para la adquisición de imágenes. Este modelo refinado es nuevamente discretizado y comparado contra imágenes reales del objeto para localizar más zonas imprecisas. Si existen más de ellas, el proceso se repite una vez más. Este proceso es repetido en sus dos fases hasta que ya no existan zonas imprecisas. Al finalizar este proceso, el modelo obtenido será preciso en todas sus zonas, bajo la precisión  $\epsilon$  exigida.

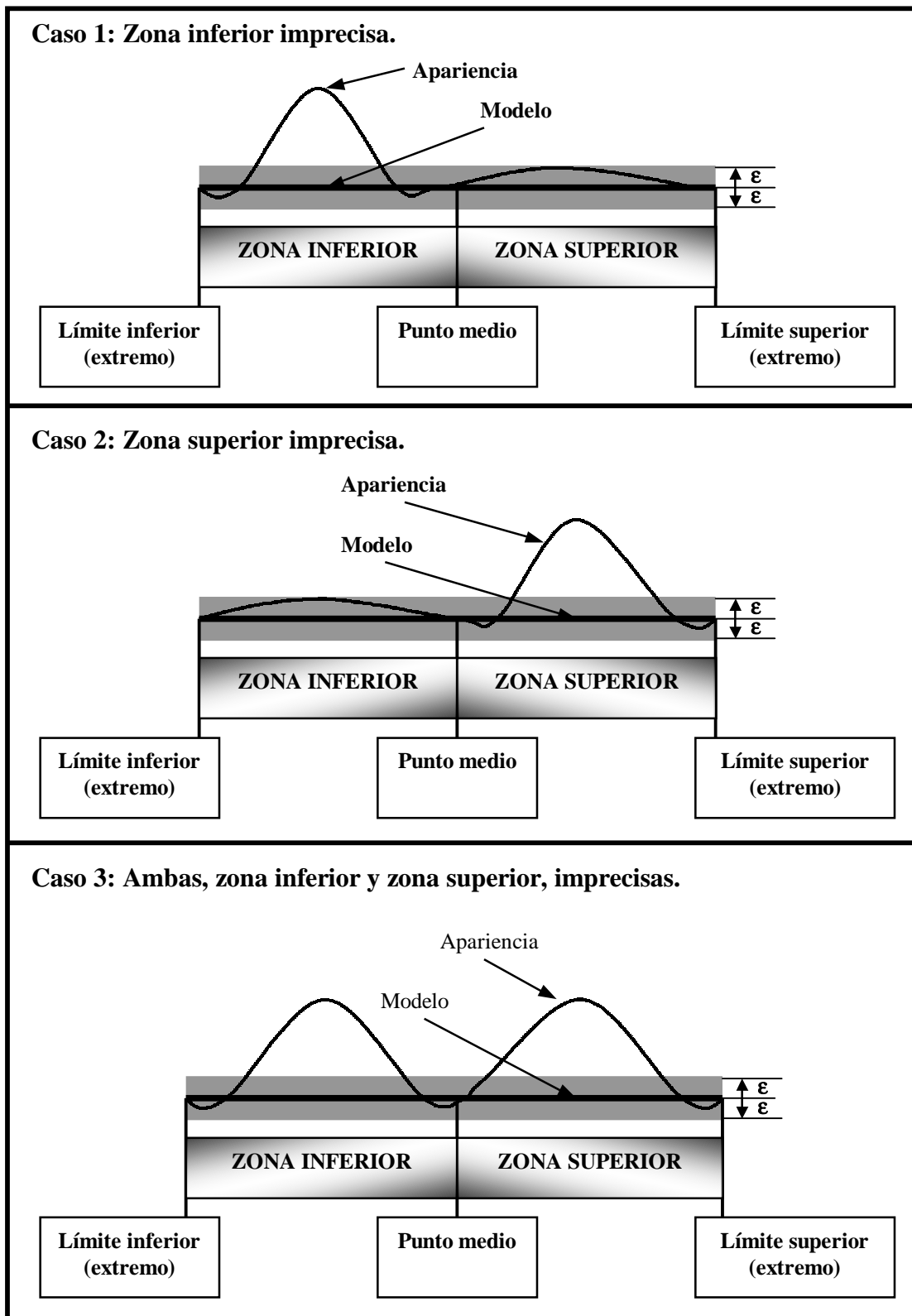


Figura 28: Los 3 casos en que se pueden presentar las zonas imprecisas.

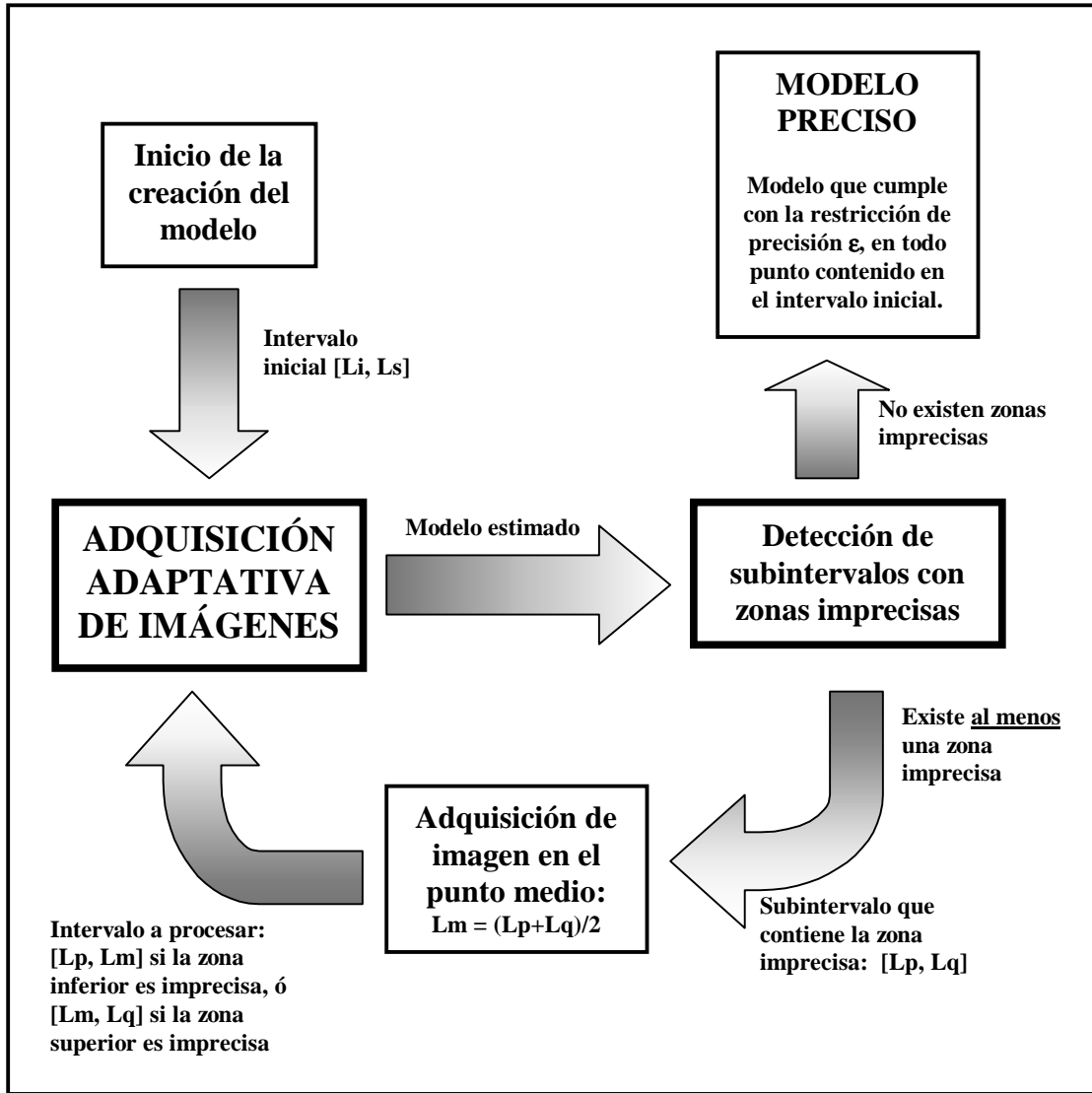


Figura 29: Diagrama de bloques que describe el proceso predictor-corrector propuesto, para la creación del modelo de un objeto, que cumple con la restricción de precisión  $\epsilon$ , en todos los puntos contenidos dentro de un intervalo inicial dado, y que por tanto, no contiene ninguna zona imprecisa.

Nótese que después de cada etapa de verificación de precisión de un modelo, el análisis de la porción del espacio de trabajo que contiene zonas imprecisas, es visto por la adquisición adaptativa como un problema en sí mismo, de estimación de imágenes requeridas para esa zona, y no como un sub-problema que depende de un problema mayor, por lo no se requiere realizar cambio alguno a la propuesta hecha en los Capítulos 4 y 5.

Nótese también, que la etapa de verificación de los modelos es requerida, para evitar que la técnica propuesta genere soluciones no factibles, en su búsqueda por un mínimo global, si se interpreta el problema de determinar cuántas y cuáles son las imágenes estrictamente necesarias, para modelar la apariencia de un objeto bajo una precisión  $\varepsilon$  preestablecida, como un problema de optimización, en donde se pretendería minimizar el número de imágenes necesarias para modelar al objeto, restringiéndose a la precisión  $\varepsilon$  exigida, y estimando además los puntos de control. Evidentemente ésta es únicamente una interpretación, pues este problema no puede ser resuelto por los métodos de optimización convencionales dada su naturaleza: no se dispone de una representación funcional, de la trayectoria de la apariencia del objeto que se quiere modelar.

Un punto importante a considerar para el proceso descrito, radica en la elección de los puntos (imágenes) que se emplearán para verificar la precisión del modelo. Estos puntos pueden provenir de un muestreo uniforme o no-uniforme del espacio de trabajo. Por simplicidad, aquí se empleará un muestreo uniforme, con un ángulo  $\theta$  de espaciamiento entre imágenes consecutivas constante. Para los resultados obtenidos en este trabajo, se limitó al hardware a no obtener imágenes consecutivas con menos de  $5^\circ$ , por lo que entonces  $\theta = 5$ . Sin embargo, si el hardware lo permite y el investigador desea mayor seguridad,  $\theta$  puede ser arbitrariamente pequeño y no necesariamente entero. Es posible emplear un muestreo no-uniforme para esta etapa, sin embargo, se requiere más investigación en esta dirección. Nótese que estas imágenes son empleadas únicamente para propósitos de verificación y son almacenadas de manera temporal por el sistema, y por tanto, una vez finalizado el proceso predictor-corrector y habiendo obtenido el modelo preciso en todas sus zonas, estas imágenes son desechadas por el sistema y, por tanto, no implican un incremento en la cantidad de imágenes requeridas para construir el modelo.

Finalmente, puede observarse que el proceso predictor-corrector propuesto, puede ser empleando para la propuesta generalizada para  $N$  parámetros, únicamente considerando que en general, la etapa de corrección dividirá una región imprecisa en  $2^N$  subregiones que deberán ser analizadas por la técnica generalizada propuesta en el Capítulo 5. Este trabajo no fue implementando en el sistema creado, por falta de tiempo, y por tanto no se ahondará más en esto. Sin embargo, nótese que tal implementación es una labor bastante simple.

### **8.1.1 Correctez y completitud del proceso propuesto**

Enseguida se probará la correctez y completitud del proceso predictor-corrector propuesto, para el caso de un parámetro (ángulo de giro de la tornamesa).

**Proposición 1:** Sea  $\varepsilon > 0$  la precisión que se le exige al proceso predictor-corrector, expuesto en la presente Sección 8.1 (Figura 29). Sea  $\theta > 0$  el mínimo ángulo de giro

que puede realizar el sistema de hardware para la adquisición de imágenes, con  $\theta$  divisor de  $360^\circ$ . Sea  $\delta > 0$  la máxima precisión proporcionada por el hardware para el objeto que se desea modelar (Ver Sección 6.1.3).

***Si  $\varepsilon \geq \delta$ , el proceso predictor-corrector siempre obtendrá al finalizar, un modelo basado en la apariencia del objeto que se está procesando, que será preciso, - bajo la precisión  $\varepsilon$  -, en todos los puntos que permite analizar el hardware, y en un número finito de pasos.***

Demostración: Supóngase que existe un punto  $p \in [0, 360]$ , para el cual el sistema de hardware puede adquirir una imagen, y que ésta no puede ser aproximada dentro de la precisión  $\varepsilon$ , con el modelo obtenido al finalizar el proceso predictor-corrector.

Sea  $l = 360/\theta$ , el número de puntos que permite analizar el hardware.  $l$  es finito debido a que  $\theta > 0$  (aunque posiblemente muy grande). Sean  $p_0, p_2, \dots, p_{l-1}$  estos  $l$  puntos. Entonces  $p = p_j$  para algún  $j \in 0, \dots, l-1$ . Sean  $p_i$  y  $p_k$  el punto inferior y superior más cercano a  $p_j$ , respectivamente, para los cuales se adquirió una imagen para construir el modelo, esto es,  $p_i \leq p_j \leq p_k$ .  $p_j$  no puede ser igual a  $p_i$  ó a  $p_k$ , porque  $SSD(p_i, p_j) = 0 < \varepsilon$  y  $SSD(p_j, p_k) = 0 < \varepsilon$ , y entonces sería bien aproximado por el modelo; por lo tanto  $p_i < p_j < p_k$ . Tampoco puede ocurrir que  $p_j$  sea el punto medio del intervalo  $[p_i, p_k]$  debido a que la adquisición adaptativa verifica que el modelo sea preciso en los puntos medios de cada intervalo que analiza. Entonces  $p_j$  está localizado en una de las zonas inferior o superior del intervalo (ver Figura 28). Esto quiere decir que se tiene alguno de los casos 1 ó 2 de la Figura 28. De acuerdo al esquema mostrado en la Figura 29, esto significa que el proceso predictor-corrector requiere adquirir una imagen más y analizar la zona imprecisa que contiene a  $p_j$ , por lo que el proceso aún no terminaría, lo cual contradice a la hipótesis. Más aún, el proceso no pudo haber terminado por haber rebasado la precisión máxima del hardware, debido a que por hipótesis  $\varepsilon \geq \delta$ . Por tanto, el modelo generado al finalizar el proceso predictor-corrector, es preciso en todos los puntos que permite verificar el hardware<sup>5</sup>. Esto prueba la precisión del modelo.

Para probar que el proceso se detiene en un número finito de pasos, nótese que las 2 etapas que componen al proceso predictor-corrector (Adquisición adaptativa y detección de zonas imprecisas) se basan en dividir en dos partes el intervalo que analizan, una y otra vez, hasta que el proceso finaliza.

Obsérvese que éste es esencialmente el comportamiento del muy conocido método de búsqueda binaria, aún cuando aquí no se están buscando datos sino aproximando la apariencia de un objeto. Debido a esto y a que la cantidad de puntos

---

<sup>5</sup> Nótese que en la práctica el punto medio debe ser redondeado al punto más cercano, válido para la precisión del hardware que se emplea, sin embargo, esto no altera el resultado obtenido, debido a que se siguen teniendo 2 zonas a verificar.

que puede emplear el modelo es  $1/2^6$ , el proceso predictor-corrector realizará a lo más  $\text{Log}_2((1/2)+1)$  divisiones de intervalos [Kr88]. Cada división implica el análisis de una zona del modelo, por lo que el algoritmo predictor-corrector analizará a lo más  $\text{Log}_2((1/2)+1)$  zonas para corregir un error, que es una cantidad finita, dado que 1 también lo es. ■

Es posible probar una proposición similar para la propuesta generalizada para N parámetros. Sin embargo y debido a que no se implementó el algoritmo predictor-corrector para el caso generalizado por falta de tiempo, no se citará aquí, y se reserva como trabajo futuro.

### 8.1.2 Resultados experimentales

El proceso predictor-corrector descrito en la Figura 29 fue incorporado al sistema para la adquisición de imágenes, descrito en el Capítulo 5, y fue empleado para corregir las imprecisiones en los modelos estimados, para los objetos de las Figuras 14 y 15. En las Tablas 10 y 11 se listan los resultados obtenidos al aplicar el proceso predictor-corrector, a los objetos de las Figuras 14 y 15, respectivamente. En el primer renglón de cada Tabla se indica si el sistema requirió corregir los modelos obtenidos en el Capítulo 5, empleando únicamente la adquisición adaptativa, para cada objeto. En el segundo renglón se muestra la cantidad de imágenes obtenida únicamente por la adquisición adaptativa de imágenes. En el tercer renglón se muestra la cantidad de imágenes obtenida al corregir el modelo. El cuarto renglón muestra el incremento requerido en la cantidad de imágenes para corregir el modelo. El quinto renglón indica el porcentaje de ahorro en imágenes, respecto a un muestreo uniforme empleando 36 imágenes, de los modelos corregidos. Finalmente, el porcentaje de ahorro promedio para todos los objetos es señalado en el último renglón.

Puede observarse que en la mayoría de los casos que así lo requirieron, la corrección se realizó con menos de 2 imágenes, y que el porcentaje de ahorro respecto a un muestreo uniforme aún continúa siendo muy significativo (en promedio, 38% y 40%, para cada grupo de objetos). Esto ilustra las ventajas del empleo de un muestreo no-uniforme y adaptativo. Aún cuando no se implementó el proceso predictor-corrector para más de un parámetro por falta de tiempo, se espera que los resultados experimentales obtenidos para un parámetro sean consistentes para el caso de N parámetros. Finalmente, en las Figuras 30 y 31 se muestra el análisis de precisión de los modelos corregidos, para los objetos de las Figuras 14 y 15, respectivamente, que así lo requirieron. Puede observarse que ahora el modelo es preciso en todas sus zonas.

---

<sup>6</sup> Recuérdese que la adquisición adaptativa de imágenes requiere que todo intervalo a analizar, contenga al menos una imagen contenida dentro del intervalo (diferente de los extremos). Esta imagen solamente es empleada para verificar la precisión del modelo, y nunca empleada para contruirlo, por lo que en un caso extremo, solamente se usarán a lo más la mitad de las imágenes que puede proporcionar el hardware (Véase Sección 4.2).



	OBJETO									
	1	2	3	4	5	6	7	8	9	10
<b>Corrección</b>	no	si	si	si	no	si	si	si	no	si
<b>Número de imágenes sin corrección</b>	23	12	4	3	27	32	24	30	26	18
<b>Número de imágenes después de corrección</b>	23	13	11	10	27	33	26	31	26	23
<b>Incremento en el número de imágenes</b>	---	1	7	7	---	1	2	1	---	5
<b>Porcentaje de ahorro (%)</b>	36	64	69	72	25	8	28	14	28	36
<b>Porcentaje de ahorro promedio:</b>										<b>38%</b>

Tabla 10: Resultados obtenidos al aplicar el proceso predictor-corrector a los objetos de la Figura 14.

	OBJETO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
<b>Corrección</b>	no	no	si	no	no	no	no	no	no	no	no	si	no	no	si	si	si	si	no	si
<b>Número de imágenes sin corrección</b>	32	27	18	26	23	18	31	25	26	22	29	11	22	26	6	13	6	26	18	9
<b>Número de imágenes después de corrección</b>	32	27	19	26	23	18	31	25	26	22	29	12	22	26	7	17	8	28	18	16
<b>Incremento en el número de imágenes</b>	---	---	1	---	---	---	---	---	---	---	---	1	---	---	1	4	2	2	---	7
<b>Porcentaje de ahorro (%)</b>	11	25	47	28	36	50	14	31	28	39	19	67	39	28	81	53	78	22	50	56
<b>Porcentaje de ahorro promedio:</b>															<b>40%</b>					

Tabla 11: Resultados obtenidos al aplicar el proceso predictor-corrector a los objetos de la Figura 15.

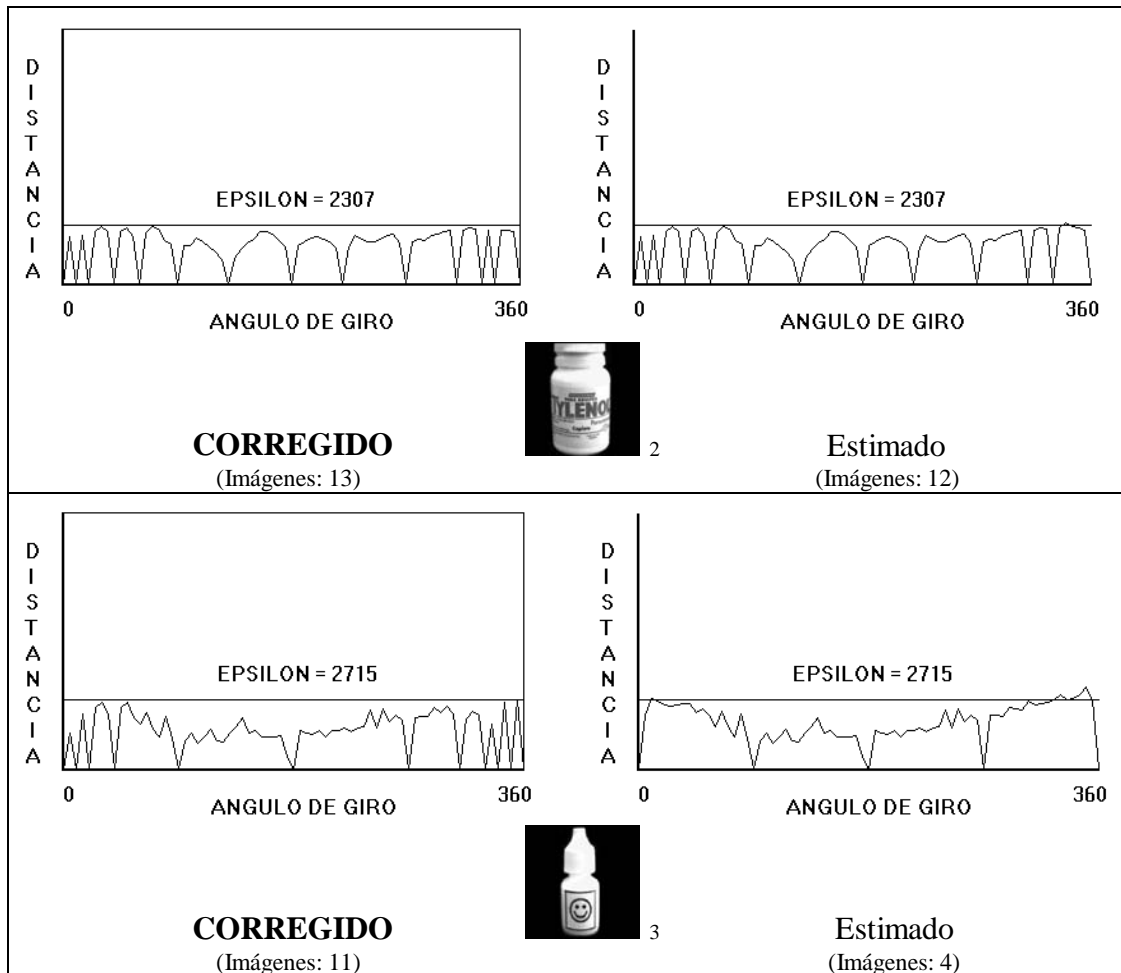


Figura 30: Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de la Figura 14, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes. Derecha: Modelo generado mediante el algoritmo 1 (Capítulo 4). Izquierda: Modelo corregido mediante el proceso predictor-corrector (Figura 29). Obsérvese la precisión en todas las zonas del modelo corregido. Solamente los objetos que requirieron corrección son mostrados.

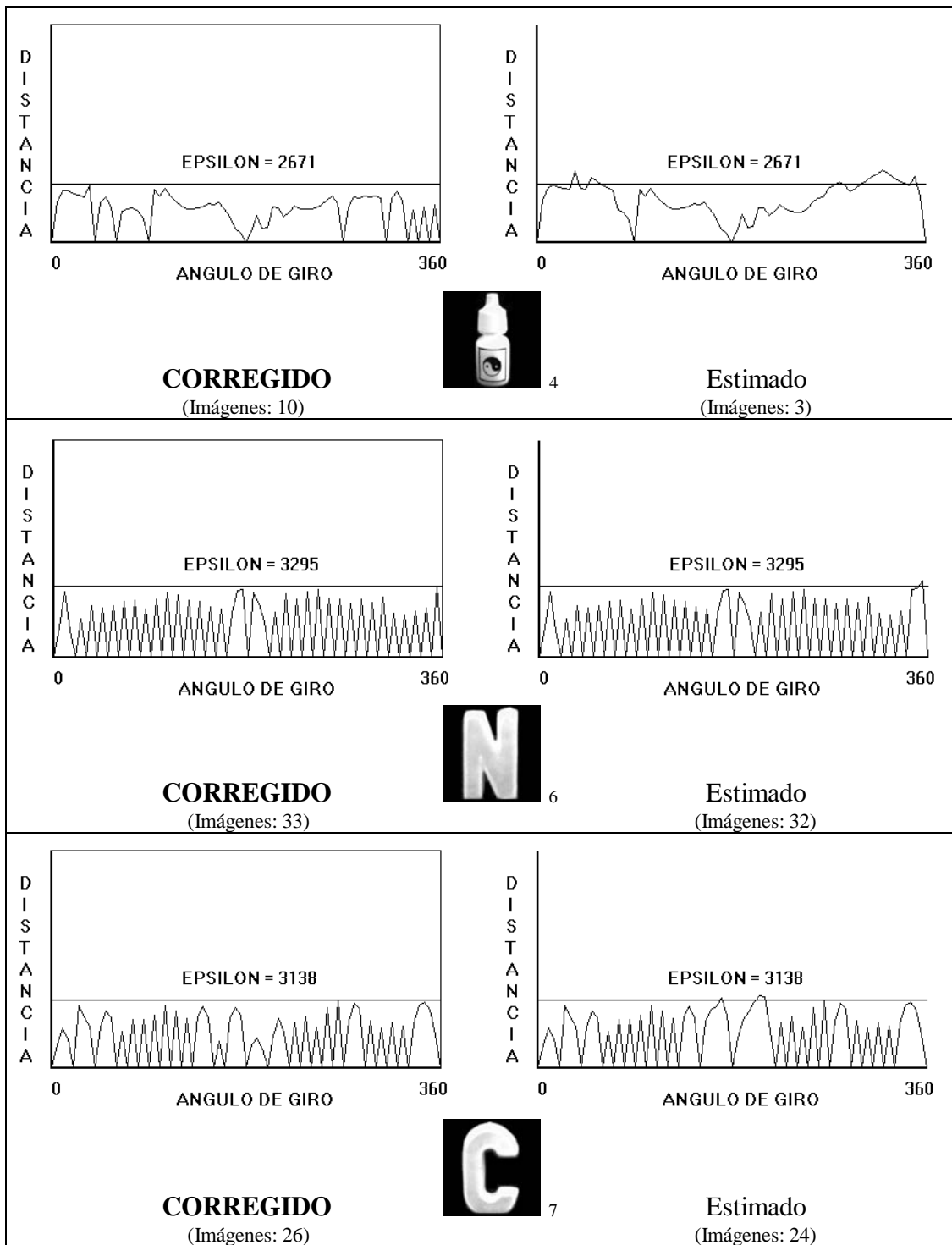


Figura 30: (continuación)

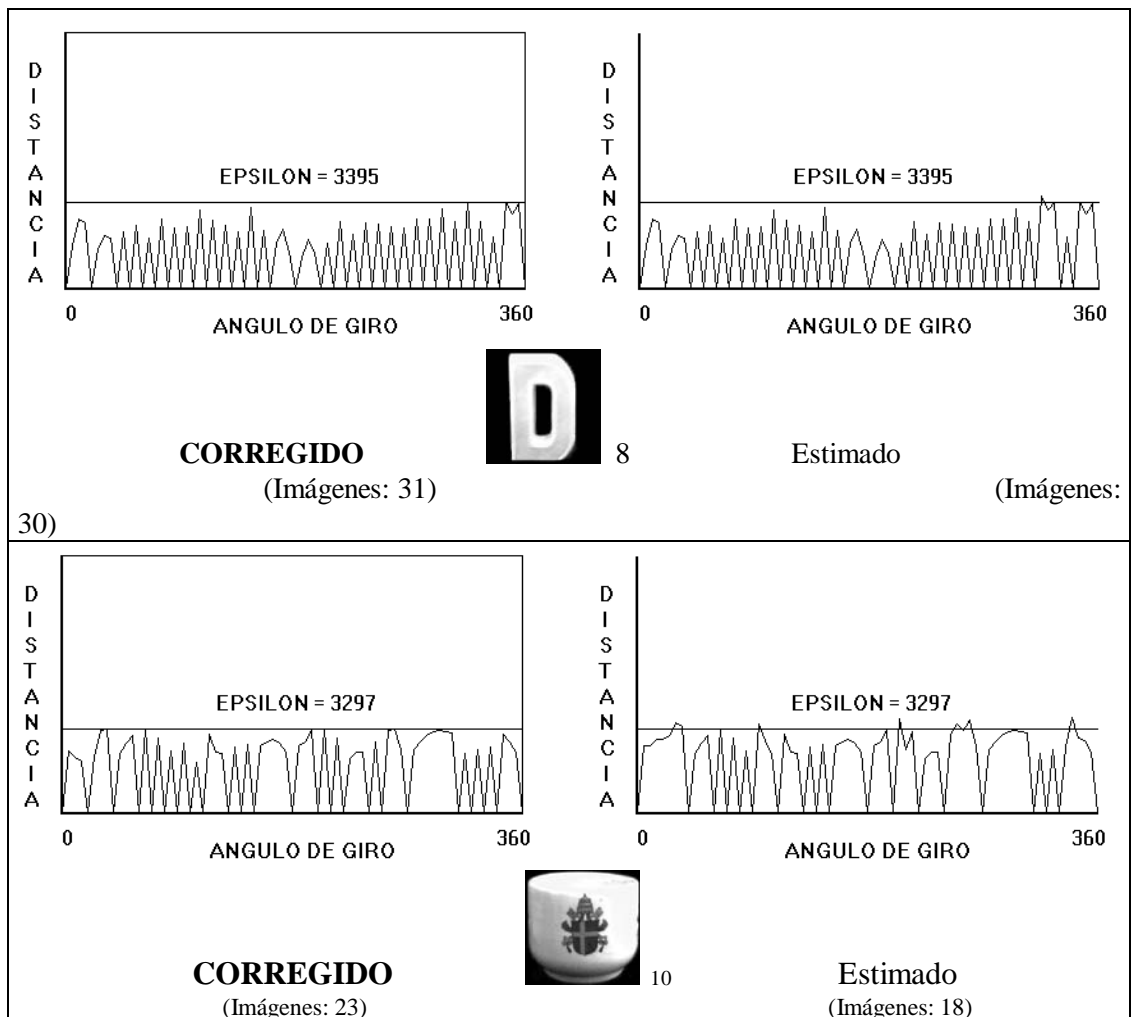


Figura 30: (continuación)

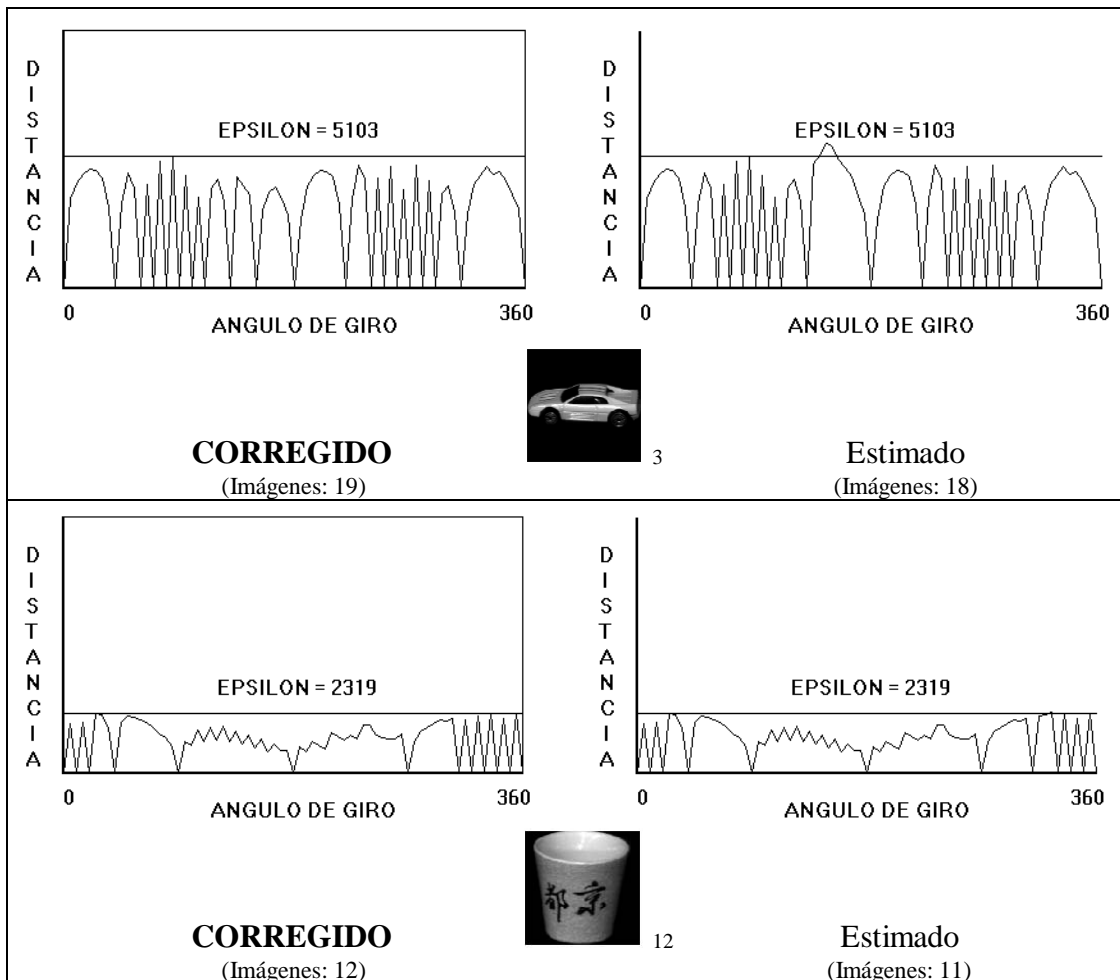


Figura 31: Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto y la apariencia real de los objetos de la Figura 15, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes. Derecha: Modelo generado mediante el algoritmo 1 (Capítulo 4). Izquierda: Modelo corregido mediante el proceso predictor-corrector (Figura 29). Obsérvese la precisión en todas las zonas del modelo corregido. Solamente los objetos que requirieron corrección son mostrados.



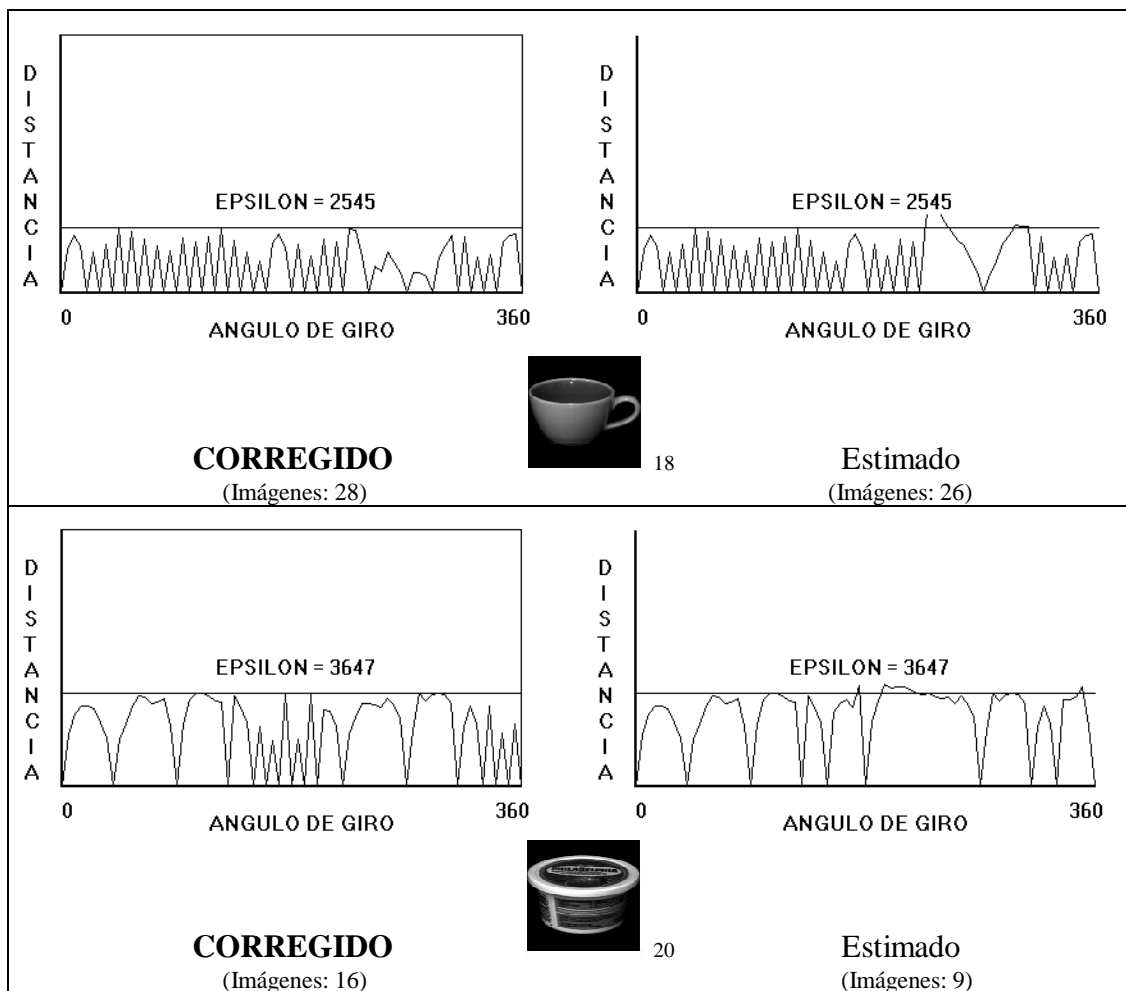


Figura 31: (continuación)

## 8.2 Determinación automática de precisión

Un problema asociado con la propuesta realizada, surge cuando al iniciar el análisis de un nuevo objeto, el sistema requiere que se le establezca la precisión  $\epsilon$  que se desea que cumpla el modelo que estimará. El valor de este parámetro deberá ser proporcionado por el usuario, dado que el sistema no cuenta con una manera de determinarlo automáticamente. Nótese que algunos de los valores posibles para este parámetro que pudiera proporcionar el usuario son inválidos, para cada objeto particular, pues harán que el sistema de software requiera que el sistema de hardware que realiza la adquisición de imágenes, supere la máxima precisión para la que fue diseñado. Esto fue mostrado en las Tablas 2, 3 y 8, en donde algunos valores (etiquetados con +) para la precisión del modelo, requerían superar la máxima precisión del hardware. Por otro lado, a partir de un determinado valor (y para

sucesivos valores mayores) para la precisión, el sistema modelará al objeto con solamente 2 imágenes (número de imágenes mínimo), lo que generará un modelo con muy baja precisión (adecuada únicamente para objetos con apariencia extremadamente simple). Debido a esto, se puede observar que los valores válidos abarcan un intervalo que es propio de cada objeto (ver Tablas 2, 3 y 8). Sin embargo, el usuario no dispone de información alguna *a priori*, para determinar el intervalo de valores válidos, y por tanto, el valor específico que deberá usar para cada objeto. En esta sección se aborda la manera en que el sistema puede determinar de manera automática, valores válidos para la precisión de los modelos, para cada objeto particular.

La solución que se propone para este problema la realizará la técnica adaptativa ya expuesta, explotando la información de las primeras k imágenes que vaya adquiriendo. Inicialmente, la precisión  $\varepsilon$  se hace igual a cero para permitir a la técnica adaptativa, iniciar la adquisición de imágenes para modelar al objeto. Después de que el sistema ha adquirido las primeras k imágenes, éstas serán consideradas como muestras estadísticas, que serán empleadas para estimar un valor válido para la precisión. Para asegurar que el valor proporcionado por el sistema sea realmente un valor válido, se deberá tomar en cuenta la precisión del hardware empleando para adquirir las imágenes.

Sean  $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k$  las primeras k imágenes adquiridas por la técnica adaptativa propuesta, adquiridas en los puntos  $p_1, p_2, \dots, p_k$  (valores para el parámetro que define la apariencia del objeto), y sea  $\theta$  la precisión máxima del hardware que adquiere las imágenes, expresada en grados ( $\theta$  será la distancia mínima a que el hardware puede adquirir imágenes consecutivas). Para cada imagen  $\mathbf{I}_i$  ( $1 \leq i \leq k$ ) se deberán adquirir dos imágenes  $\mathbf{X}_i$  y  $\mathbf{Y}_i$  en las posiciones  $p_i - \theta$  y  $p_i + \theta$ , respectivamente. Enseguida se usará la expresión (5) para interpolar linealmente entre  $\mathbf{X}_i$  y  $\mathbf{Y}_i$  y obtener:

$$g_i(\lambda) = \lambda \mathbf{X}_i + (1-\lambda) \mathbf{Y}_i ; \quad 0 \leq \lambda \leq 1, \quad 1 \leq i \leq k \quad (9)$$

finalmente, la expresión (9) será evaluada en  $\lambda=0.5$ ,  $g(0.5)$ , y el valor resultante comparado con  $\mathbf{I}_i$  mediante SSD (ver Definición 5), para obtener:

$$\beta_i = \text{SSD}(\mathbf{I}_i, g(0.5)) \quad 1 \leq i \leq k \quad (10)$$

nótese que  $\beta_i$  ( $1 \leq i \leq k$ ) tiene como objetivo indicar con qué precisión se puede aproximar la apariencia del objeto en el intervalo  $[p_i - \theta, p_i + \theta, ]$  empleando la máxima precisión del hardware. Evidentemente, entre mayor sea el valor de k, se tendrá información de un número mayor de porciones de la apariencia del objeto.

Los k valores obtenidos mediante (10) pueden ser procesados de varias maneras, para determinar un valor para la precisión  $\varepsilon$  en el modelo. Aquí únicamente se exponen 2 de ellas:

$$\text{i) } \varepsilon = \left( \sum_1^k \beta_i \right) / k \quad (11)$$

$$\text{ii) } \varepsilon = \text{máx}(\beta_1, \beta_2, \dots, \beta_k) \quad (12)$$



si se consideran igualmente probables todos los valores de la precisión, dentro del intervalo de valores permitidos para un determinado objeto, la primer opción aproxima el valor esperado de la precisión  $\epsilon$  dentro del intervalo de análisis. Por otro lado, la segunda opción aproxima a la precisión máxima, lograda con un muestreo a  $\theta$  grados de incremento, como se expuso en la Sección 6.1.3.

Evidentemente, en ambas expresiones (11) y (12), la estimación será mejor si se dispone de un mayor número de muestras. Sin embargo, si el número de muestras es relativamente grande, la cantidad de imágenes adicionales (que son desechadas posteriormente por el sistema, una vez determinado el valor de  $\epsilon$ ) será también grande. Por tanto, el investigador deberá determinar cuántas muestras deberá emplear el sistema. Finalmente, nótese que si el número de muestras es pequeño, el sistema podrá determinar en ambas expresiones (11) y (12), un valor para  $\epsilon$  fuera del rango permitido, y que superará la precisión del hardware, por lo que es deseable calcular las expresiones (11) ó (12) en base a una precisión del hardware  $2\theta$  en lugar de  $\theta$ , para superar este problema.

### 8.2.1 Resultados experimentales

En las Tablas 12 y 13 se muestran los resultados obtenidos al aplicar las 2 estrategias propuestas para determinar el valor de la precisión  $\epsilon$ , sobre los objetos de las Figuras 14 y 15, para diferentes valores para el número de muestras  $k$  (3, 7 y 15 muestras), y empleando  $\theta=5$  como la máxima precisión del hardware. Los cálculos se hicieron empleando  $2\theta$  en lugar de  $\theta$ , para no rebasar la precisión del hardware. Los valores en negritas denotan resultados fuera del rango permitido para cada objeto.

OBJETO	INTERVALO	PROMEDIO			MÁXIMO		
	VÁLIDO	3	7	15	3	7	15
1	6405 - 8000	6559	<b>6164</b>	<b>6078</b>	7737	7737	8505
2	2307 - 3500	<b>1555</b>	<b>1721</b>	<b>1768</b>	<b>1701</b>	<b>2236</b>	2491
3	2715 - 3000	<b>1596</b>	<b>1675</b>	<b>1748</b>	<b>2014</b>	<b>2492</b>	<b>2639</b>
4	2671 - 3000	<b>1628</b>	<b>1440</b>	<b>1487</b>	<b>2106</b>	<b>2106</b>	<b>2106</b>
5	4992 - 8000	<b>4614</b>	<b>4430</b>	<b>4674</b>	5440	5440	6617
6	3295 - 8000	3573	3660	3728	4461	4461	4461
7	3138 - 8000	<b>2739</b>	<b>2833</b>	<b>2831</b>	3855	3855	3855
8	3395 - 8000	<b>3184</b>	<b>3381</b>	3492	4090	4090	4280
9	4201 - 8000	<b>3790</b>	<b>4159</b>	<b>4108</b>	<b>4170</b>	4695	5476
10	3297 - 4000	<b>2604</b>	<b>2549</b>	<b>2618</b>	<b>2808</b>	<b>2808</b>	<b>3223</b>

Tabla 12: Resultados obtenidos al aplicar las 2 estrategias propuestas, para calcular automáticamente un valor para la precisión  $\epsilon$  con el cual, el sistema construirá el modelo de cada uno de los objetos de la Figura 14.

De los resultados listados en ambas Tablas 12 y 13, se puede observar que, en general, los mejores resultados son obtenidos con la segunda estrategia. Esto era de esperarse debido a que tal estrategia está enfocada a no rebasar la precisión del hardware. Por supuesto, los mejores resultados son obtenidos con un número mayor de muestras, sin embargo, recuérdese que existen objetos que no requieren tantas imágenes para ser modelados. Aún cuando en ambas Tablas 12 y 13 se puede observar que ninguna estrategia es infalible, bajo ninguno de los valores para el número de muestras listado, un número de muestras igual a 7, puede ser apropiado para casi cualquier objeto, empleando la estrategia de el máximo valor (segunda estrategia). Este valor servirá como referencia para el investigador, para darle una idea aproximada de cuál es el rango de valores permitidos para cada objeto.

Aún cuando los valores determinados automáticamente por el sistema, pueden ser empleados sin cambio alguno, la precisión adecuada deberá ser estipulado por el investigador, en base al tipo de reconocimiento que el sistema realizará. Si el reconocimiento será extra-clase (es decir, entre objetos muy distintos entre sí), es aconsejable emplear una precisión inferior a la sugerida por el sistema; por el contrario, si el reconocimiento será intra-clase (es decir, entre objetos muy parecidos entre si), una precisión igual o superior a la sugerida por el sistema deberá ser una elección razonable.

OBJETO	INTERVALO VÁLIDO	PROMEDIO			MÁXIMO		
		3	7	15	3	7	15
1	2073 - 8000	2420	2618	2556	3287	3392	3392
2	3514 - 8000	3963	3890	3753	5421	5421	5421
3	5103 - 8000	5475	<b>4691</b>	<b>4446</b>	6728	6728	6728
4	2945 - 8000	<b>2378</b>	<b>2815</b>	2972	3037	4178	4462
5	5252 - 8000	5706	<b>5173</b>	<b>5053</b>	6786	6786	7073
6	7282 - 8000	<b>6303</b>	<b>5199</b>	<b>5084</b>	7923	7923	7923
7	2360 - 7500	2777	2422	2440	2993	2993	2993
8	2152 - 7000	<b>2091</b>	<b>2022</b>	<b>2048</b>	2826	2826	2826
9	5817 - 8500	7377	6213	6128	8139	8139	8139
10	3287 - 8000	<b>2555</b>	<b>2817</b>	<b>2912</b>	<b>3191</b>	3477	3796
11	2509 - 8000	3146	2848	2851	3752	3752	3756
12	2319 - 3000	<b>835</b>	<b>1285</b>	<b>1282</b>	<b>1335</b>	2380	2445
13	4164 - 8000	<b>3049</b>	<b>3396</b>	<b>3539</b>	<b>3789</b>	<b>3981</b>	5411
14	3448 - 4500	4106	<b>3425</b>	<b>3414</b>	4859	4859	4859
15	1445 - 2500	<b>761</b>	<b>698</b>	<b>771</b>	<b>808</b>	<b>808</b>	<b>1309</b>
16	1267 - 2000	<b>548</b>	<b>663</b>	<b>752</b>	<b>602</b>	<b>1125</b>	1387
17	2733 - 4000	<b>1491</b>	<b>1388</b>	<b>1485</b>	<b>1559</b>	<b>1715</b>	2337
18	2545 - 5500	<b>2048</b>	<b>2432</b>	<b>2497</b>	2775	3487	3487
19	4655 - 8000	4707	<b>3995</b>	<b>3861</b>	5868	5868	5868
20	3647 - 5500	<b>2924</b>	<b>2851</b>	<b>2859</b>	<b>3277</b>	<b>3396</b>	<b>3396</b>

Tabla 13: Resultados obtenidos al aplicar las 2 estrategias propuestas, para calcular automáticamente un valor para la precisión  $\epsilon$  con el cual, el sistema construirá el modelo de cada uno de los objetos de la Figura 15.

Por último, una variante de la segunda estrategia (máximo valor), garantiza que siempre el valor sugerido por el sistema será soportado por el hardware: si el número de muestras es igual a la mitad del máximo número máximo de puntos que puede proporcionar el hardware, se obtendrá la precisión máxima que puede proporcionar el hardware, y que ha sido empleada a lo largo de este trabajo de tesis. Sin embargo, nótese que esta estrategia requiere analizar una cantidad mayor de puntos (36 en este trabajo de tesis), que los que en general requiere el modelo del objeto.

### 8.3 Empleo de interpolación no-lineal en la propuesta

Una interrogante que surge al analizar la propuesta presentada en esta tesis, radica en si podría ser empleado un esquema de interpolación no-lineal, en lugar del lineal (y el N-lineal) que ha sido estudiado, y que resultados y mejoras se obtendría de ello.

Un conjunto de esquemas de interpolación que sería un buen candidato para tal sustitución, lo constituye el conjunto de polinomios, dada la gran cantidad de estudios sobre sus propiedades, que pueden ser encontradas en la literatura. Sin embargo, una de estas propiedades establece que el empleo de polinomios de un grado elevado (superior a 3) para realizar interpolación es, en general, una mala estrategia, debido a la naturaleza oscilante de los polinomios. Esta propiedad es compartida también por otros esquemas de interpolación no-lineal. Otro punto más que hace no deseable el empleo de polinomios de grado superior a uno, es la cantidad de cálculos, que crece según el grado del polinomio. Sin embargo, con polinomios de grado inferior a 3, esta cantidad parece aún razonable.

A la par del estudio realizado en esta tesis con esquemas de interpolación lineal, se realizó un ensayo empleando dos esquemas de interpolación no-lineal: interpolación cuadrática e interpolación cúbica, ésta última mediante el empleo de *splines*. Tales esquemas fueron incorporados a la técnica propuesta, y estudiado brevemente su comportamiento y resultados.

En el caso del empleo de polinomios cuadráticos, se modificó el paso 7 del algoritmo 1 mostrado en el Capítulo 4, para incluir tal forma de no-linealidad. En este caso, se requiere considerar tres imágenes  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$  para definir el polinomio cuadrático a emplear para la interpolación, y mediante el empleo de la expresión:

$$g(\lambda) = 2(\lambda-0.5)(\lambda-1)\mathbf{X} - 4\lambda(\lambda-1)\mathbf{Y} + 2\lambda(\lambda-0.5)\mathbf{Z} \quad ; \quad 0 \leq \lambda \leq 1 \quad (13)$$

el algoritmo 1 fue capaz de emplear cuadráticas para aproximar el modelo del objeto. A diferencia del caso lineal, en el caso cuadrático, el algoritmo debe verificar la buena aproximación en 2 puntos:  $g(0.25)$  y  $g(0.75)$ . Ambos puntos deben ser bien aproximados por el modelo que se está construyendo; de otra forma, el algoritmo

divide el intervalo que está analizando en 2 subintervalos para analizarlos por separado.

Objeto	Máxima precisión		Imágenes requeridas	
	Lineal	Cuadrática	Lineal	Cuadrática
1	6405	9282	23	18
2	2307	2552	12	14
3	2715	2553	4	10
4	2671	2551	3	10
5	4992	7001	27	16
6	3295	4053	32	28
7	3138	4011	24	22
8	3395	4556	30	22
9	4201	4735	26	26
10	3297	3578	18	18

Tabla 14: Comparativo entre interpolación lineal y no-lineal respecto a la máxima precisión obtenida y el número de imágenes requeridas para modelar a los objetos de la Figura 14.

Objeto	Máxima precisión		Imágenes requeridas	
	Lineal	Cuadrática	Lineal	Cuadrática
1	2073	3225	32	22
2	3514	4957	27	22
3	5103	7150	18	12
4	2945	4176	26	24
5	5252	7339	23	18
6	7282	7998	18	16
7	2360	2778	31	30
8	2152	3106	25	22
9	5817	8098	26	22
10	3287	3988	22	22
11	2509	3914	29	18
12	2319	2734	11	10
13	4164	4619	22	22
14	3448	4557	26	26
15	1445	1100	6	16
16	1267	1454	13	14
17	2733	2100	6	12
18	2545	2882	26	28
19	4655	6420	18	14
20	3647	3811	9	12

Tabla 15: Comparativo entre interpolación lineal y no-lineal respecto a la máxima precisión obtenida y el número de imágenes requeridas para modelar a los objetos de la Figura 15.

Los resultados experimentales para el caso cuadrático son ilustrados en las Tablas 14 y 15 para los objetos de las Figuras 14 y 15, respectivamente. Si se comparan estos resultados con los obtenidos mediante interpolación lineal mostrados en las Tablas 4 y 5, respectivamente, se puede observar que aún cuando para el caso cuadrático la cantidad de imágenes requeridas es menor respecto al caso lineal, para todos los objetos en general, la máxima precisión en el caso no-lineal es muy superior que en el caso lineal. En las Figuras 32 y 33 se grafica la distancia entre los modelos generados mediante cuadráticas y la apariencia del objeto, para los objetos mostrados en las Figuras 14 y 15, respectivamente. Comparando estas Figuras 32 y 33 contra sus correspondientes Figuras 19 y 20, respectivamente, se puede observar que los modelos generados mediante interpoladores cuadráticos son menos precisos que los correspondientes generados mediante interpoladores lineales. Esto se debe a que para definir una cuadrática se requieren 3 puntos (uno más que para polinomios lineales), y es más difícil para el sistema ajustar 3 puntos a solamente 2 (lineal). Más aún, la interpolación no-lineal no obtiene ventaja de las zonas de la apariencia, donde el comportamiento es cuasi-lineal. Finalmente, el tiempo de ejecución para cuadráticas fue superior al correspondiente empleado por lineales. Esto se debe a la mayor cantidad de operaciones entre imágenes que el esquema requiere realizar.

En el caso de *splines*, se siguió una estrategia distinta: se obtuvo una aproximación lineal de la apariencia con la técnica propuesta, y posteriormente fue suavizado el modelo obtenido, mediante el empleo de *splines*. Los resultados obtenidos generaron un modelo suave de la apariencia del objeto; sin embargo, las imprecisiones fueron extremadamente elevadas. Esto se debió a la naturaleza oscilante de los polinomios de grado 3 que se emplearon, y que si bien generan un modelo suave de la apariencia, no se sujetan a la restricción de precisión  $\epsilon$ . Evidentemente, una mejor estrategia sería derivar las expresiones para calcular un *spline* que se sujete a las restricciones de precisión exigidas, sin embargo, esto se reserva para un trabajo futuro, al igual que un estudio más profundo de la posibilidad de incorporar interpolación no-lineal a la presente propuesta de tesis.

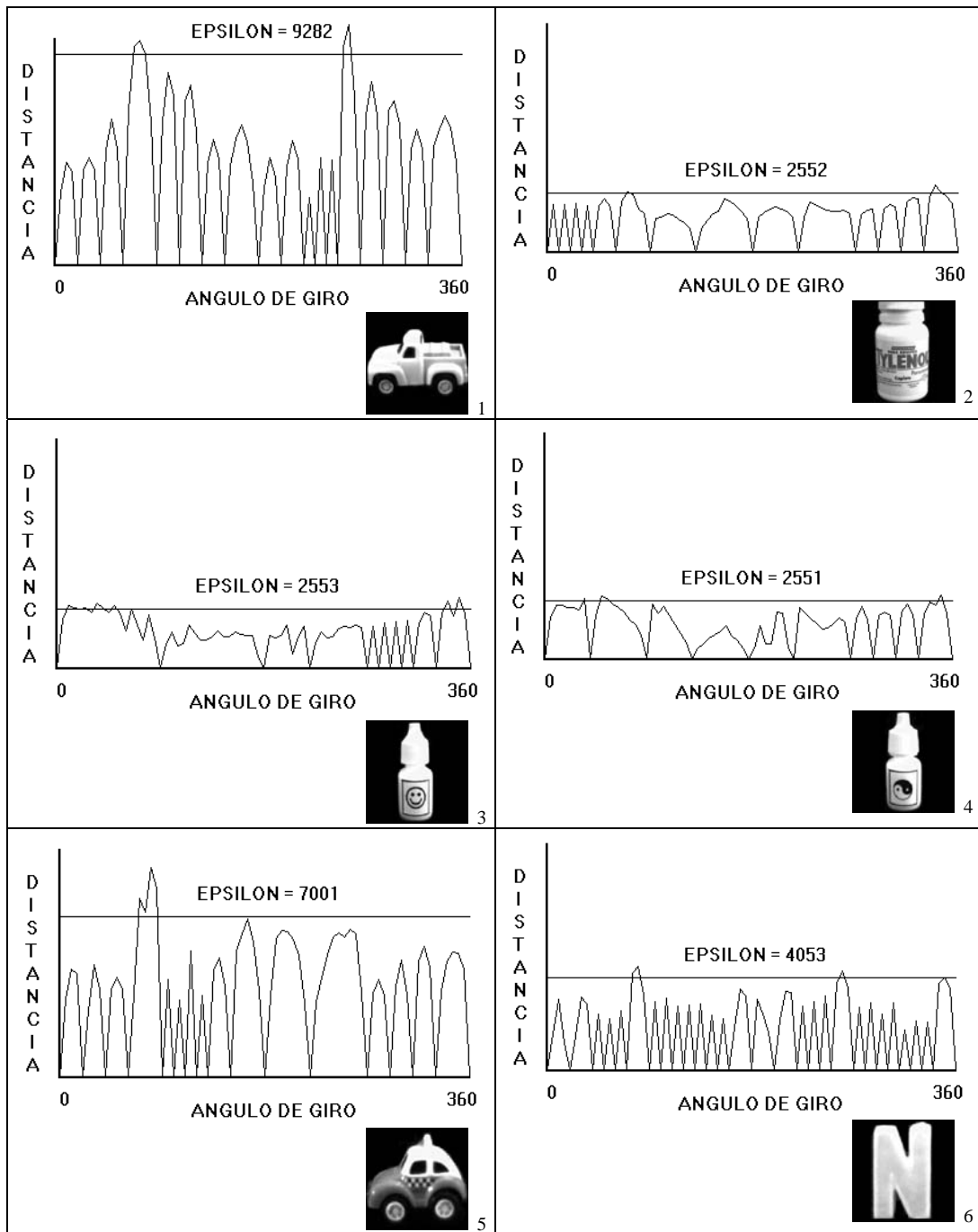


Figura 32: Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto empleando no-linealidad y la apariencia real de los objetos de la Figura 14, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.

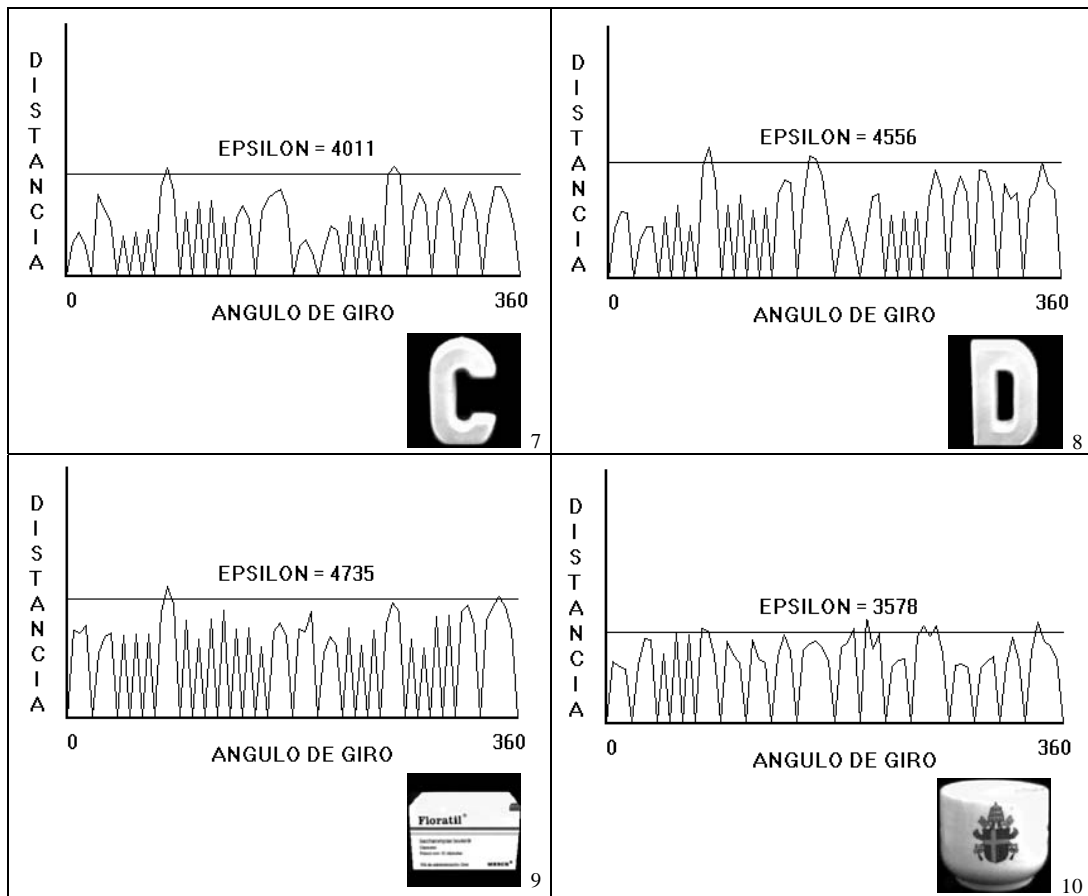


Figura 32: (continuación).

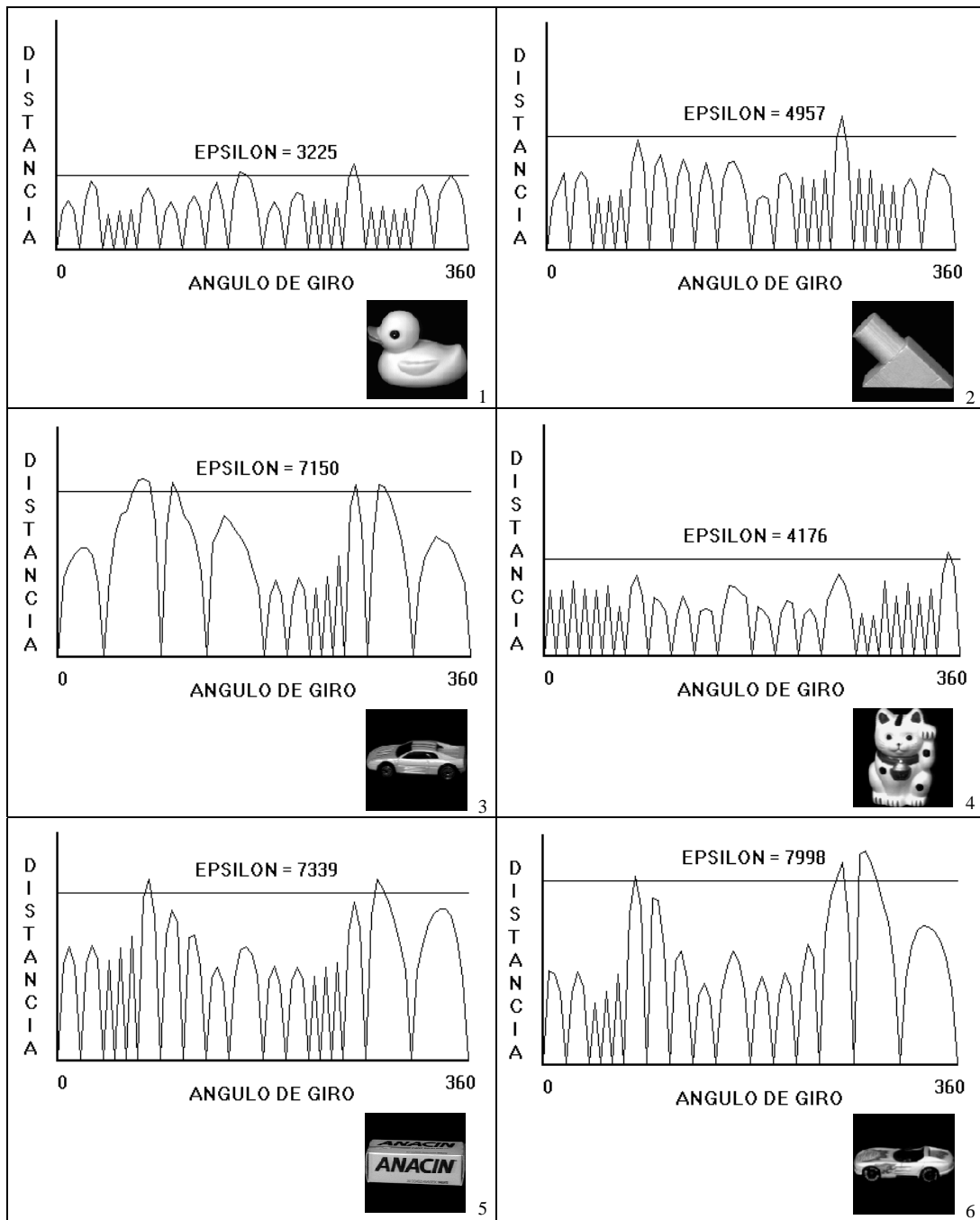


Figura 33: Graficado de la SSD (eje Y) entre el modelo estimado por el algoritmo propuesto empleando no-linealidad y la apariencia real de los objetos de la Figura 15, empleando un parámetro (rotación de la tornamesa, eje X), en base a la precisión  $\epsilon$  señalada. Las partes del gráfico bajo la precisión  $\epsilon$  denotan buenas aproximaciones, y las partes sobre la precisión  $\epsilon$  indican que se requiere mejorar el modelo del objeto en esas partes.



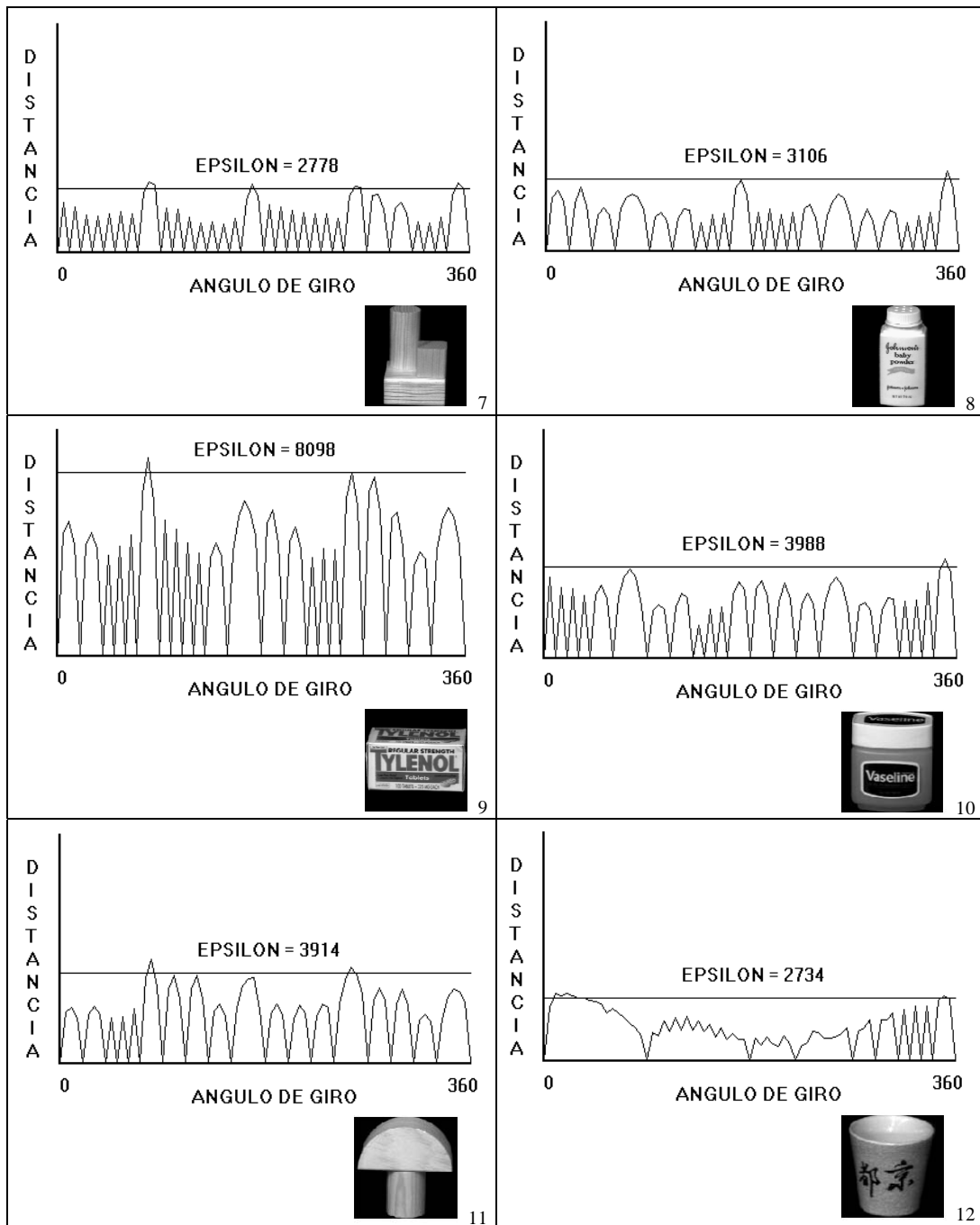


Figura 33: (continuación).

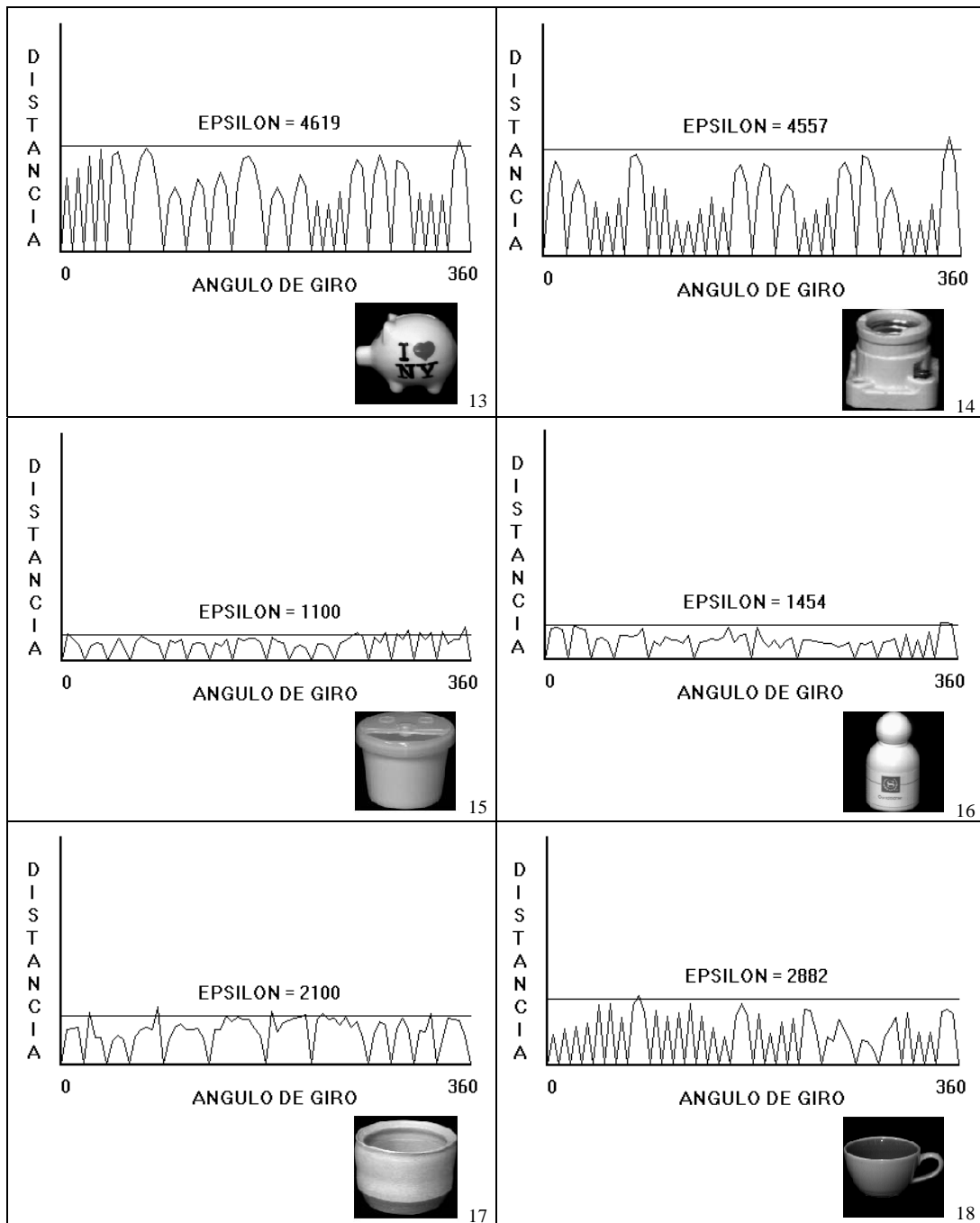


Figura 33: (continuación).

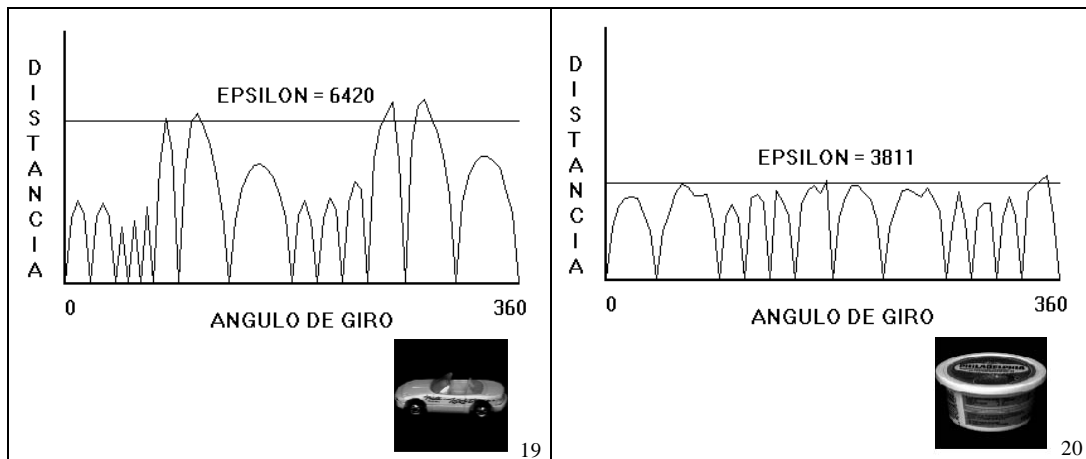


Figura 33: (continuación).

#### 8.4 Selección de un enfoque (¿geométrico ó basado en apariencia?)

En la introducción de este trabajo de tesis, fueron comentadas las ventajas y desventajas tanto de los enfoques geométricos, como de los basados en apariencia. Sin embargo, un aspecto que no ha sido aún abordado en esta tesis, se refiere a cuál sería el criterio para seleccionar alguno de los dos enfoques, para una aplicación específica.

Debido a que el enfoque basado en apariencia se basa en la suposición de que los objetos serán reconocidos en condiciones similares, a las presentes en el momento de modelar al objeto (posición, iluminación, etc.), toda aplicación en donde esta suposición sea válida es un campo fértil para emplear dicho enfoque. Esto incluye por supuesto, algunas aplicaciones de tipo industrial, como la citada en el Capítulo 7.

Nótese que el enfoque basado en apariencia logra altos índices de reconocimiento, siempre y cuando la suposición previa se cumpla. Sin embargo, éste no es el caso de muchas otras aplicaciones, en donde, por supuesto, se deberá preferir el empleo de enfoques geométricos, donde es posible el empleo de características invariantes ó semi-invariantes, que no requieren sujetarse a la restricción que limita a los enfoques basados en apariencia. Sin embargo, este criterio deberá adoptarse con mucho cuidado, pues recuérdese de capítulos previos, que si se define la apariencia de un objeto mediante múltiples parámetros, entonces los enfoques basados en apariencia pueden trabajar bajo condiciones variables de iluminación, tanto o mejor aún que los enfoques geométricos.

Por tanto, aunado al criterio anterior, se deberá considerar como otro criterio de selección el nivel de detalle requerido para determinada aplicación: si el reconocimiento de los objetos puede realizarse con pocas características de los

objetos, y éstas características son fácilmente modelables con geometría (como es el caso del reconocimiento óptico de caracteres, la visión aérea, etc.), se debiera preferir un enfoque geométrico; si por el contrario, considerando pocas características se producen ambigüedades en el proceso de reconocimiento (como es el caso de objetos muy similares entre sí, objetos con la misma forma geométrica, etc.), se debería pensar en el empleo de un enfoque basado en apariencia. Nótese además, que en algunas aplicaciones se puede explotar un enfoque híbrido (como por ejemplo, en el reconocimiento de rostros).

Otro posible criterio de selección a considerar, se refiere al tipo de material que forma al objeto. Debido a que la apariencia es fuertemente influenciada por las condiciones en que se realiza la adquisición de imágenes, los enfoques basados en apariencia no serán capaces de reconocer objetos translúcidos en escenas arbitrarias, como por ejemplo, objetos de cristal o plástico transparente, debido a que un fondo distinto al empleado en la etapa de modelado impedirá el reconocimiento del objeto. Existen algunas excepciones, como por ejemplo el trabajo de Nelson y Selinger [Ne98b] que al restringirse al contorno del objeto pueden abordar este problema. Sin embargo, en general, en este caso se debiera preferir un enfoque geométrico que sea capaz de abordar este problema específico.

Un último criterio de selección que debiera ser empleado para elegir un enfoque para una determinada aplicación, radica en la facilidad o complejidad de modelar los objetos de interés bajo un enfoque particular. En efecto, debido a que los enfoques basados en apariencia son capaces de modelar de una forma fácil, objetos arbitrariamente complejos en su apariencia (geometría, textura, color, etc.), se deberán preferir tales enfoques en aplicaciones donde la geometría del objeto es difícil de manejar (como por ejemplo con algunos objetos de tipo industrial, como los estudiados en el Capítulo 7). Nótese que desde este punto de vista, los enfoques basados en apariencia deben ser preferidos sobre los enfoques geométricos, dada su gran facilidad y simplicidad para modelar y reconocer objetos.

Finalmente, nótese que ambos enfoques pueden fallar en muchos casos (como por ejemplo, bajo condiciones pobres de iluminación, ocultamiento parcial, etc.), por lo que se espera que en un futuro cercano ambos enfoques se complementen entre sí [P&H96].

# CAPÍTULO 9

## ***CONCLUSIONES, APORTACIONES Y TRABAJO FUTURO***

---

El reconocimiento de objetos usando modelos basados en apariencia es un enfoque propuesto en la década de los 90's. Este enfoque permite reconocer objetos arbitrariamente complejos en su geometría y/o textura, y por tanto presenta ventajas sobre aquellos enfoques que requieren el empleo explícito de la geometría de los objetos.

Para el modelado y reconocimiento basados en apariencia, la técnica hoy conocida como basada en espacios propios (*eigenspaces*) ha mostrado grandes potencialidades. Tal técnica permite modelar los objetos mediante una representación compacta, al emplear el Análisis de Componentes Principales (PCA de sus siglas en inglés). Esta técnica requiere la presencia simultánea de una gran cantidad de imágenes para modelar a los objetos. Esto trae como consecuencia que el sistema que modela a los objetos requiera una gran cantidad de espacio de almacenamiento para las imágenes, y por tanto, que requiera una gran cantidad de tiempo de cómputo para modelar a los objetos.

En este trabajo doctoral se planteó la posibilidad de reducir la cantidad de imágenes requerida para la construcción de los modelos de los objetos. Esta reducción se realizó mediante el empleo de un muestreo no-uniforme de imágenes. Tal muestreo fue realizado mediante una técnica adaptativa propuesta, para la adquisición de las imágenes estrictamente necesarias para construir los modelos de los objetos. A la par, tal técnica permitió mejorar la precisión de tales modelos.

La técnica se basa en el empleo de interpolación lineal y SSD para determinar las imágenes estrictamente necesarias para capturar la apariencia de un objeto, dentro de un margen de error  $\epsilon$  preestablecido.

Un proceso predictor-corrector fue propuesto como complemento a la técnica adaptativa, para garantizar la precisión de los modelos en todas las zonas, que el sistema de hardware para la adquisición de imágenes permita verificar. Así mismo, la correctez y completitud de tal proceso fueron demostradas.

Mediante diversos experimentos, se pudo corroborar la efectividad de la técnica propuesta, al implementarse en un sistema de software que fue aplicado sobre varios tipos de objetos. En particular, fue empleada para analizar la precisión de la librería de imágenes de objetos de la Universidad de Columbia, COIL-20.

Finalmente, se mostró de qué manera es posible acoplar la técnica propuesta a la técnica de espacios propios, mostrando que la precisión garantizada por el algoritmo propuesto es preservada después de construir el modelo del objeto en el espacio propio.

El acoplamiento de la técnica propuesta no es exclusivo para la técnica de espacios propios, aunque en la presente tesis solamente fue experimentado para esta técnica. Es posible acoplarla a las técnicas que emplean Redes Neuronales, *Wavelets*, etc., mediante un análisis para cada caso. Actualmente se está investigando la posibilidad de su empleo bajo el enfoque de Redes Neuronales.

## **Aportaciones**

Las contribuciones principales de esta propuesta son:

- Se propuso el empleo de un muestreo no-uniforme de imágenes, como forma de reducir la cantidad de imágenes requeridas para modelar un objeto cualquiera.
- Se propuso una técnica adaptativa para la adquisición de imágenes no-uniforme, mediante la cual se pueden obtener modelos de cualquier objeto, tan precisos como se requieran, dentro de las limitaciones del hardware para la adquisición de imágenes.
- Mediante un proceso predictor-corrector propuesto, la precisión del modelo obtenido por la técnica es garantizada.
- La técnica se adapta de acuerdo al comportamiento de la apariencia de cada objeto, y por lo tanto,
- La técnica permite una reducción de la cantidad de imágenes para construir los modelos de los objetos (imágenes innecesarias son eliminadas). Como consecuencia:
- Se ha logrado una reducción en el tiempo de cómputo requerido para construir los modelos.
- Fue realizado un análisis de las ventajas de usar muestreo no-uniforme.
- Finalmente, se mostró cómo la técnica puede ser acoplada a la técnica de espacios propios, para mejorar el reconocimiento de objetos y para lograr una reducción en el tiempo de procesamiento de los modelos.

## Limitaciones

El presente trabajo de tesis presenta inevitablemente limitaciones; algunas de ellas son:

- La cantidad de imágenes requeridas para modelar un objeto es reducida, respecto a un muestreo uniforme, siempre y cuando la estimación de posición del objeto no sea importante. Si por el contrario, la estimación de posición es importante, tal reducción podría no ser tan relevante. La presente propuesta no aborda este problema, aún cuando se obtuvieron buenos resultados para algunos objetos, y aún cuando los algoritmos propuestos son fácilmente modificables para lograr una estimación correcta de posición de los objetos, como se indica en el trabajo futuro.
- El empleo de otras métricas distintas a la utilizada no fue abordado, principalmente por falta de tiempo.

## Trabajo futuro

Indudablemente, la presente propuesta es susceptible de mejoras, que podrían investigarse en un futuro cercano. Algunas de ellas pueden ser:

- La presente propuesta se ha limitado, (sobre todo por factores de tiempo), al empleo de solamente esquemas de interpolación lineal para aproximar la apariencia de los objetos, aún cuando se investigó brevemente un esquema de interpolación no-lineal. Sin embargo, la exploración de esquemas no-lineales podrían ser un campo fértil que merece ser explorado adecuadamente.
- La presente propuesta podrá ser empleada para la estimación de posición del objeto mediante una modificación simple: bastará incluir un paso que verifique la buena aproximación dentro del espacio paramétrico de las imágenes interpoladas, y no solamente en el espacio de apariencia, como fue propuesto en la presente tesis.
- La presente propuesta no aborda la posibilidad de explotar las simetrías que presenten los objetos. Evidentemente si se considera este aspecto, se podría obtener una reducción aún mayor en la cantidad de imágenes requeridas.
- La presente propuesta es factible de ser modificada para funcionar con otras métricas distintas a la empleada.
- En la presente propuesta se ha planteado el empleo conjunto del muestreo no-uniforme para la adquisición de imágenes, con la técnica de espacios propios. Sin embargo, investigación adicional es necesaria para fusionarlo con otras técnicas como son las redes neuronales, *wavelets*, etc.
- Un aspecto interesante a investigar en el futuro, consistiría en el empleo de la Teoría estadística de la información, para determinar límites teóricos para las

cantidades mínima y máxima de imágenes requeridas para modelar a los objetos, y comparar estas cotas con los resultados obtenidos por la presente propuesta.

- Finalmente, dado que los enfoques basados en apariencia no se limitan solamente al empleo de imágenes captadas mediante luz visible, un tópico más ha investigar en el futuro, será determinar la aplicación de la presente propuesta, sobre imágenes de luz ultravioleta, infraroja, etc.



# APENDICE A

## *PUBLICACIONES*

---

### **En revistas con arbitraje:**

- “Non-Uniform Sampling For Improved Appearance-Based Models”, Altamirano, L.C., Altamirano, L., Alvarado, M., en revisión, Pattern Recognition Letters, The Netherlands, 2002.
- “Adaptive Image Acquisition Guided by Interpolation for Automatic Generation of Appearance Based Models”, Altamirano, L.C., Altamirano, L., Alvarado, M., en revisión, Computación y Sistemas, México, 2002.
- “Automatic generation of N-parametric models of objects, using information of their appearance, with applications to object recognition“, Altamirano, L.C., Altamirano, L., Alvarado, M., en preparación, 2002.

### **En Congresos con arbitraje:**

- “Predicción y corrección de modelos basados en apariencia, con aplicaciones al reconocimiento de objetos ”, Altamirano, L.C., Altamirano, L., Alvarado, M., en preparación, 2002.
- “Generación automática de modelos N-paramétricos de objetos, empleando información de su apariencia“, Luis Carlos Altamirano, Leopoldo Altamirano, Matías Alvarado, memorias del VI Taller Iberoamericano de Reconocimiento de Patrones, TIARP2001, Noviembre 2001.
- “Adquisición Adaptativa de Imágenes Guiada por Interpolación, para la Generación Automática de Modelos Basados en Apariencia”, Altamirano, L.C., Altamirano, L., Alvarado, M., memorias del Congreso Internacional CIC2000, Noviembre 2000.

- **“Reconocimiento de Objetos usando Modelos Basados en Apariencia”, Altamirano, L.C., Altamirano, L., Alvarado, M., Congreso Virtual Cómputo2000, UNAM, Octubre 2000 (por invitación).**
- **“Reconocimiento de Objetos Articulado Usando Modelos Basados en Apariencia de Bajo Costo”, Altamirano, L.C., Altamirano, L., Alvarado, M., Taller de Inteligencia Artificial TAINA’99, Octubre 1999.**

### **Informes Técnicos:**

- **“Automatic generation of N-parametric models of objects, using information of their appearance, with applications to object recognition“, Altamirano, L.C., Altamirano, L., Alvarado, M., por aparecer, 2002.**
- 
- **“Non-Uniform Sampling for Better Appearance-Based Models”, Altamirano, L.C., Altamirano, L., Alvarado, M., Informe Técnico No. 84, Serie Roja, CIC-IPN, Agosto 2000.**
- **“Adaptive Image Acquisition Guided by Interpolation for Automatic Generation of Appearance Based Models”, Altamirano, L.C., Altamirano, L., Alvarado, M., Informe Técnico No. 83, Serie Roja, CIC-IPN, Agosto 2000.**
- **“Reconocimiento de Objetos usando Modelos Basados en Apariencia”, Altamirano, L.C., Altamirano, L., Alvarado, M., Informe Técnico No. 45, Serie Azul, CIC-IPN, Agosto 1999.**

# APENDICE B

## *PROYECTOS DE INVESTIGACIÓN*

---

El presente trabajo doctoral, se ha realizado bajo el marco de los siguientes proyectos de investigación. Todos ellos están aprobados y apoyados por la institución que se señala, y actualmente se encuentran finalizados. Fueron realizados por equipos de trabajo tanto del Centro de Investigación en Computación del IPN, como del Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE):

➤ CONACYT

“ Un sistema multiagentes para la elaboración de modelos y su reconocimiento, con información de su apariencia “. Ref. 131991A.  
Año: 2000.

➤ REDII

“ Un sistema multiagentes para la elaboración de modelos y su reconocimiento, con información de su apariencia “. Año: 2000-2001.

➤ CGPI

“ Sistema multiagentes para la elaboración de modelos de objetos y su reconocimiento, usando información de su apariencia “. Año: 2000.

# APENDICE C

## ALGORITMO GENERALIZADO PARA N PARÁMETROS

---

Descripción: Algoritmo generalizado para construir el modelo basado en la apariencia de un objeto, definida ésta mediante N parámetros.

Parámetros de entrada:

- N es el número de parámetros que definen la apariencia del objeto.
- L es la matriz de  $2 \times N$  que contiene los límites inferiores y superiores, que limitan la zona a analizar, para todas las N dimensiones. El primer renglón  $L[1][1..N]$  contiene los N límites inferiores y el segundo renglón  $L[2][1..N]$  los N límites superiores. Por ejemplo, el elemento  $L[1][1]$  contiene el límite inferior de la primer dimensión, y el elemento  $L[2][1]$  contiene el límite superior de la primer dimensión.
- $\epsilon$  es la precisión deseada para el modelo.

Salida:

- Imágenes estrictamente requeridas para construir el modelo del objeto, para una precisión  $\epsilon$ .
- 

1. Para  $i[1] = 1$  a 2 hacer  
    Para  $i[2] = 1$  a 2 hacer  
        .....  
        Para  $i[N] = 1$  a 2 hacer  
            Adquirir imagen en la posición (  $L[i[1]][1], L[i[2]][2], \dots, L[i[N]][N]$  )  
            { Se adquieren las  $2^N$  imágenes en los vértices de la zona a analizar }  
2. Para  $k = 1$  a N hacer  
     $L_M[k] = (L[2][k] + L[1][k]) / 2$   
    { Calcula el punto medio de la zona a analizar }  
3. Adquirir imagen (R) en la posición: (  $L_M[1], L_M[2], \dots, L_M[N]$  ).  
4. Interpolarse N-linealmente entre las  $2^N$  imágenes adquiridas en el paso 1, para obtener la función N-lineal interpolante  $f(t_1, t_2, \dots, t_N), 0 \leq t_i \leq 1$ .  
5. Si es posible interpolar la imagen (R) usando  $f(0.5, 0.5, \dots, 0.5)$  dentro de un margen de error  $\epsilon$ , usando SSD como criterio de comparación, esto es:  
     $\| R - f(0.5, 0.5, \dots, 0.5) \|^2 \leq \epsilon$   
    ir al paso 8.

6. Para  $k = 1$  a  $N$  hacer
    - $\text{aux}[k] = (\text{L}[2][k] - \text{L}[1][k]) / 2$
    - $\text{temp}[1] = \text{L}[1][1]$
    - Para  $i[1] = 1$  a 2 hacer
      - {  $\text{temp}[2] = \text{L}[1][2]$
      - Para  $i[2] = 1$  a 2 hacer
        - {  $\text{temp}[3] = \text{L}[1][3]$
        - Para  $i[3] = 1$  a 2 hacer
        - .....
        - Para  $i[N-1] = 1$  a 2 hacer
          - {  $\text{temp}[N] = \text{L}[1][N]$
          - Para  $i[N] = 1$  a 2 hacer
            - { empilar ( $\text{temp}[1], \dots, \text{temp}[N], \text{temp}[1] + \text{aux}[1], \dots, \text{temp}[N] + \text{aux}[N]$ )
            - $\text{temp}[N] = \text{temp}[N] + \text{aux}[N]$
            - }
          - $\text{temp}[N-1] = \text{temp}[N-1] + \text{aux}[N-1]$
          - }
    - .....
    - $\text{temp}[2] = \text{temp}[2] + \text{aux}[2]$
    - }
    - $\text{temp}[1] = \text{temp}[1] + \text{aux}[1]$
    - }
    - { *Se envían a la pila  $2^N$  subregiones en que se divide la región que se está analizando, para su posterior análisis* }
  7. Ir al paso 9.
  8. Guardar las  $2^N$  imágenes adquiridas en el paso 1, como imágenes necesarias para construir el modelo del objeto.
  9. Si la pila no está vacía, entonces desempilar una zona a analizar, asignarla a ( $\text{L}[1][N], \text{L}[2][N]$ ) y regresar al paso 1.
  10. FIN.
-

# REFERENCIAS BIBLIOGRÁFICAS

- AI96 Altamirano, L., Eckstein, W., "The importance of feature visibility for the evaluation of a matching hypothesis", Computer Analysis of Images and Patterns, 13th. International Conference on Pattern Recognition (ICPR'96), August, 23-30, Vienna, Austria, 1996.
- AI99a Altamirano, L.C., Altamirano, L., Alvarado, M., "Reconocimiento de Objetos usando Modelos Basados en Apariencia", Informe Técnico No. 45, Serie Azul, CIC-IPN, Agosto 1999.
- AI99b Altamirano, L.C., Altamirano, L., Alvarado, M., "Reconocimiento de Objetos Articulados Usando Modelos Basados en Apariencia de Bajo Costo", Taller de Inteligencia Artificial TAINA'99, Octubre 1999.
- AI00a Altamirano, L.C., Altamirano, L., Alvarado, M., "Adaptive Image Acquisition Guided by Interpolation for Automatic Generation of Appearance Based Models", Informe Técnico No. 83, Serie Roja, CIC-IPN, Agosto 2000.
- AI00b Altamirano, L.C., Altamirano, L., Alvarado, M., "Non-Uniform Sampling for Improved Appearance-Based Models", Informe Técnico No. 84, Serie Roja, CIC-IPN, Agosto 2000.
- AI00c Altamirano, L.C., Altamirano, L., Alvarado, M., "Reconocimiento de Objetos usando Modelos Basados en Apariencia", Congreso Virtual Cómputo2000, UNAM, Octubre 2000.
- AI00d Altamirano, L.C., Altamirano, L., Alvarado, M., "Adquisición Adaptativa de Imágenes Guiada por Interpolación, para la Generación Automática de Modelos Basados en Apariencia", memorias del Congreso Internacional CIC2000, Noviembre 2000.
- AI01 Altamirano, L.C., Altamirano, L., Alvarado, M., "Generación automática de modelos N-paramétricos de objetos, empleando información de su apariencia", memorias del VI Taller Iberoamericano de Reconocimiento de Patrones, TIARP2001, Noviembre 2001.
- AI02a Altamirano, L.C., Altamirano, L., Alvarado, M., "Adaptive Image Acquisition Guided by Interpolation for Automatic Generation of Appearance Based Models", por aparecer en Computación y Sistemas, 2002.

- Al02b Altamirano, L.C., Altamirano, L., Alvarado, M., "Non-Uniform Sampling For Improved Appearance-Based Models", por aparecer en Pattern Recognition Letters, 2002.
- Al02c "Automatic generation of N-parametric models of objects, using information of their appearance, with applications to object recognition", Altamirano, L.C., Altamirano, L., Alvarado, M., en preparación.
- Ba95 Baker, S., Nayar, S.K., "A Theory of Pattern Rejection", Columbia University Technical Report, CUCS-013-95, May 1995.
- Ba96 Baker, S., Nayar, S.K., "A Theory of Pattern Rejection", Proceedings of ARPA Image Understanding Workshop, San Francisco, February 1996.
- Be97 Belhumeur, Peter N., Hespanha, Joao P., Kriegman, David J., "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 711-720, July 1997.
- Be98a Belhumeur, Peter N., Kriegman, David J., "What is the Set of Images of an Object Under All Possible Illumination Conditions ?", International Journal of Computer Vision, Vol. no. 28, Issue No. 3, 245-260, 1998.
- Be98b Belhumeur, P., Jacobs, D., "Comparing Images Under Variable Illumination", to appear in the Proceedings of CVPR98, 1998.
- Br95 Bregler, C., Omohundro, S., "Nonlinear Image Interpolation using Manifold Learning, NIPS-7, MIT Press, 1995.
- Bu85 Burden, R., and Faires, D., "Numerical Analysis", Third Edition, PWS, Boston, E.E.U.A., 1985.
- Bu97 Burge, M., Burger, W., "Learning Visual Ideals", Proceedings of Image Analysis and Processing. 9th. International Conference, ICIAP '97, Florence Italy, Semptember 1997. Vol. II, pp. 316-323.
- By98 Byne, J., Anderson, J., "Surface Based Hypothesis Verification in Intensity Images Using Geometric and Appearance Data", Proceedings of Computer Vision ACCV'98, Third Asian Conference on Computer Vision, Hong Kong, China, January 1998, Vol. I, pp. 177-184.

- Co00 Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J., "Coupled-View Active Appearance Models", The Eleventh British Machine Vision Conference, University of Bristol, September 2000.
- Ep94 Epstein, R., Hallinan, P., Yuille, A., " $5 \pm 2$  Eigenimages suffice: An empirical investigation of low-dimensional lighting models. Technical Report 94-11, Harvard University, 1994.
- Ep96 Epstein, R., Yuille, A.L., Belhumeur, P.N., "Learning object representations from lighting variations", Object Representation in Computer Vision II, in Proceedings of ECCV'96 International Workshop, Cambridge U.K., April 1996, 179-200.
- Fa88 Farin, G., "Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide", Academic Press, Inc., 1988.
- Ge98 Georghiades, A., Kriegman, D., Belhumeur, P., "Illumination Cones for Recognition Under Variable Lighting: Faces", to appear in the Proceedings of CVPR98, 1998.
- Go96 González, R., Woods, R., "Tratamiento digital de imágenes", Addison-Wesley/Diaz de Santos, 1996.
- Go98 Gong, S., Ong, E., Loft, P., "Appearance-Based Face Recognition under Large Head Rotations in Depth", Proceedings of Computer Vision ACCV'98, Third Asian Conference on Computer Vision, Hong Kong, China, January 1998, Vol. II, pp. 679-686.
- Gu68 Guzmán, A., "Computer Recognition of Three-Dimensional Objects in a Visual Scene", MIT AI Laboratory Technical Report 228, December 1968.
- He94 Hel-Or, Y., Werman, M., "Model Based Pose Estimation of Articulated and Constrained Objects", Proceedings of Third European Conference on Computer Vision, Stockholm, Sweden, May 1994.
- He95 Hebert, M., Ponce, J., Boult, T., Gross, A., "Report on the 1995 Workshop on 3-D Object Representations in Computer Vision", in Proceedings of Object Representation in Computer Vision, International NSF-ARPA Workshop, New York City, NY, USA, December 1994.
- Ho87 Horn, B., "Robot Vision", The MIT Press, Third printing, 1987.



- Ho95 Horaud, R., Sossa, H., "Polyhedral object recognition by indexing", Pattern Recognition, Vol. 28, No. 12, 1855-1870, 1995.
- Hu98 Hu, Wei., Seales, W., "Recognition in Wavelet-Compressed Imagery", Proceedings of Computer Vision ACCV'98, Third Asian Conference on Computer Vision, Hong Kong, China, January 1998, Vol. I, pp. 362-369.
- Iw98 Iwai, Y., Hata, T., Yachida, M., "Gesture Recognition from Image Motion Based on Subspace Method and HMM", Proceedings of Computer Vision ACCV'98, Third Asian Conference on Computer Vision, Hong Kong, China, January 1998, Vol. II, pp. 639-646.
- Jo98 Jones, M., Poggio, T., "Multidimensional Morphable Models", Proc. ICCV'98, Bombay, India, 1998.
- La95 Lanitis, A., Taylor, C., Cootes, T., "A Unified Approach to Coding and Interpreting Face Images", Proc. ICCV'95, Cambridge, M.A., 1995.
- Le00 Leonardis, A., Bischof, H., "Robust Recognition Using Eigenimages", Computer Vision and Image Understanding, 78, 99-118, 2000.
- Ma76 Marsden, J.E., and Tromba, A.J., "Vector Calculus", W.H. Freeman and Company, Eds., 1976.
- Ma97 Martin, J., Crowley, J., "An Appearance-Based Approach to Gesture-Recognition", Proceedings of Image Analysis and Processing. 9th. International Conference, ICIAP '97, Florence Italy, September 1997. Vol. II, pp. 340-347.
- Me92 Munkelt, O., "Reconocimiento de objetos en imágenes de video por medio de árboles de aspectos" (en alemán), Disertación, Technische Universität München, 1992.
- Mo96 Moghaddam, B., Pentland, A., "Probabilistic Visual Learning for Object Representation", in Early Visual Learning, Edited by Shree K. Nayar and Tomaso Poggio, New York Oxford, Oxford University Press, 1996.
- Mo98 Moghaddam, B., Wahid, W., Pentland, A., "Beyond Eigenfaces: Probabilistic Matching for Face Recognition", International Conference on Automatic Face & Gesture Recognition, Nara, Japan, April 1998.

- Mo00 Mokhtarian, F., and Abbasi, S., "Automatic Selection of Optimal Views in Multi-view Object Recognition", The Eleventh British Machine Vision Conference, University of Bristol, September 2000.
- Mu92 Murase, H., Nayar, S.K., "Visual Learning of Object Models from Appearance, Technical Report, Columbia University, CUCS-054-92.
- Mu95 Murase, H., Nayar, S.K., "Visual learning and recognition of 3-D objects from appearance", International Journal of Computer Vision, Vol. 14, No. 1, 5-24, January 1995.
- Mu96 Mundy, J., Liu, A., Pillow, N., Zisserman, A., Abdallah, S., Utcke, S., Nayar, S., Rothwell, C., "An experimental comparison of appearance and geometric model based recognition", Object Representation in Computer Vision II, in Proceedings of ECCV'96 International Workshop, Cambridge U.K., April 1996, 247-269.
- Na94a Nayar, S.K., Murase, H., Nene, S.A., "Learning, Positioning, and Tracking Visual Appearance," Proceedings of IEEE International Conference on Robotics and Automation (ICRA 94), San Diego, May 1994.
- Na94b Nayar, S.K., Nene, S.A., Murase, H., "Real-Time 100 Object Recognition System," Technical Report, Columbia University, CUCS-019-95, September 1994.
- Na95a Nayar, S.K., Baker, S., Murase, H., "Parametric Feature Detection", Technical Report, Columbia University, CUCS-028-95, October 1995.
- Na95b Nayar, S.K., Murase, H., "Dimensionality of illumination manifolds in appearance matching", Technical Report, Columbia University, CUCS-021-94, August 1994, Revised: September 1995.
- Na95c Nayar, S.K., Nene, S.A., Murase, H., "Subspace Methods for Robot Vision", Technical Report, Columbia University, CUCS-06-95, March 1995.
- Na96a Nayar, S., Murase, H., Nene, S., "Parametric Appearance Representation", in Early Visual Learning, Edited by Shree K. Nayar and Tomaso Poggio, New York Oxford, Oxford University Press, 1996.
- Na96b Nayar, S.K., Nene, S.A., Murase, H., "Real-Time 100 Object Recognition System," Proceedings of ARPA Image Understanding Workshop, San Francisco, February 1996.

- Na96c Nayar, S.K., Nene, S.A., Murase, H., "Subspace Methods for Robot Vision", IEEE Transactions on Robotics and Automation, to appear in 1996.
- Na96d Nayar, S.K., Murase, H., "Dimensionality of illumination manifolds in appearance matching", Object Representation in Computer Vision II, in Proceedings of ECCV'96 International Workshop, Cambridge U.K., April 1996, 165-178.
- Na96e Nayar, S.K., Baker, S., Murase, H., "Parametric Feature Detection", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 96), San Francisco, June 1996.
- Ne94a Nene, S.A., Nayar, S.K., Murase, H., "SLAM: A Software Library for Appearance Matching," Technical Report, Columbia University, CUCS-019-94, November 1994.
- Ne94b Nene, S.A., Nayar, S.K., Murase, H., "Software Library for Appearance Matching (SLAM)", Proceedings of ARPA Image Understanding Workshop, Monterey, November 1994.
- Ne95 Nene, S.A., Nayar, S.K., "A Simple Algorithm for Nearest Neighbor Search in High Dimensions", Technical Report, Columbia University, CUCS-030-95, October 1995.
- Ne96 Nene, S.A., Nayar, S.K., "Closest Point Search in High Dimensions," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 96), San Francisco, June 1996.
- Ne97 Nelson, Randal C., Selinger, A., "Experiments on (Intelligent) Brute Force Methods for Appearance-Based Object Recognition", DARPA Image Understanding Workshop, 1197-1205, 1997.
- Ne98a Nelson, Randal C., Selinger, A., "Large-Scale tests of a Keyed, Appearance-Based 3-D Object Recognition System". To appear 1998.
- Ne98b Nelson, Randal C., Selinger, A., "A Cubist approach to Object Recognition". To appear 1998.
- Ne98c Nelson, Randal C., Selinger, A., "A Perceptual Grouping Hierarchy for Appearance-Based 3D Object Recognition". To appear 1998.

- Oh96 Ohba, K., Ikeuchi, K., "Recognition of the Multi Specularity Objects using the Eigen-Window", Proceedings of International Conference on Pattern Recognition, August 1996.
- Oh98 Ohba, K., Sato, Y., Ikeuchi, K., "Appearance Based Visual Learning and Object Recognition with Illumination Invariance", Proceedings of Computer Vision ACCV'98, Third Asian Conference on Computer Vision, Hong Kong, China, January 1998, Vol. II, pp. 424-431.
- Oj83 Oja, E., "Subspace Methods of Pattern Recognition", Res. Studies Press, Hertfordshire, 1983.
- P&H96 Ponce, J., Hebert, M., Zisserman, A., "Report on the 1996 International Workshop on Object Representation in Computer Vision", Object Representation in Computer Vision II, in Proceedings of ECCV'96 International Workshop, Cambridge U.K., April 1996, 1-8.
- P&L96a Pope, A., Lowe, D., "Learning Probabilistic Appearance Models for Object Recognition", in Early Visual Learning, Edited by Shree K. Nayar and Tomaso Poggio, New York Oxford, Oxford University Press, 1996.
- P&L96b Pope, A.R., Lowe, D.G., "Learning appearance models for object recognition", Object Representation in Computer Vision II, in Proceedings of ECCV'96 International Workshop, Cambridge U.K., April 1996, 201-220.
- Pa95 Pauli, J., Benkwitz, M., Sommer, G., "RBF Networks Appearance-Based Object Detection", Proceedings of ICANN, Paris, Volume 1, pp. 359-364, 1995.
- Pa96 Pauls, J., Bricolo, E., Logothetis, N., "View Invariant Representations in Monkey Temporal Cortex: Position, Scale and Rotational Invariance", in Early Visual Learning, Edited by Shree K. Nayar and Tomaso Poggio, New York Oxford, Oxford University Press, 1996.
- Pe94 Pentland, A., Moghaddam, B., Starner, T., "View-Based and Modular Eigenspaces for Face Recognition", IEEE Conference on Computer Vision & Pattern Recognition, 1994, also Technical Report No. 245, MIT Media Laboratory Perceptual Computing Section.

- Po94 Pope, A. R., "Model-Based Object Recognition - A Survey of Recent Research", Technical Report 94-04, University of British Columbia, January 1994.
- Po96 Poggio, T., Beymer, D., "Regularization Networks for Visual Learning", in Early Visual Learning, Edited by Shree K. Nayar and Tomaso Poggio, New York Oxford, Oxford University Press, 1996.
- Po97 Pontil, M., Verri, A., "Direct Aspect-Based 3-D Object Recognition", Proceedings of Image Analysis and Processing. 9th. International Conference, ICIAP '97, Florence Italy, Semptember 1997. Vol. II, pp. 300-307.
- Se96 Seales, B., Yuan, C., Hu, W., Cutts, M., "Content Analysis of Compressed Video", University of Kentucky Computer Science Departament Technical Report No. 2, 65-96, August 1996.
- Se98 Seales, W., Yuan, C., "Improved Image Classification Using Morphing", Proceedings of Computer Vision ACCV'98, Third Asian Conference on Computer Vision, Hong Kong, China, January 1998, Vol. II, pp. 233-240.
- Sch97 Schmid, C., and Mohr, R., "Local grayvalue invariants for image retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5):530-535, 1997.
- Sch00 Schiele, B., and Crowley, J.L., "Recognition without Correspondence using Multidimensional Receptive Field Histograms", International Journal of Computer Vision, 36(1), 31-50, 2000.
- Sh87 Shirai, Y., "Three-Dimensional Computer Vision", Springer-Verlang Ed. 1987.
- Sh94 Shapiro, L.G., Costa, M.S., "Appearance-Based 3D Object Recognition", Object Representation in Computer Vision, Vol. I, in Proceedings of International NSF-ARPA Workshop, New York City, NY, USA, December 1994. 51-64.
- Sh99 Shimon, E., "Representation and Recognition in Vision", The MIT Press, Cambridge, Massachusetts, London, England, 1999.
- St00 Stevens, M.R., Beveridge, J.R., "Integrating graphics and vision for object recognition", Kluwer Academic Publishers, 2000.

- Su97 Surendro, K., Anzai, Y., "Non-rigid Object Recognition Using Principal Component Analysis and Geometric Hashing", Proceedings of 7th. International Conference of Computer Analysis of Images and Patterns. Kiel, Germany, Sept. 1997.
- Tu91 Turk, M., Pentland, A., "Eigenfaces for Recognition", J. Cognitive Neuroscience, Vol. 3, no. 1, 1991.