



FACULTAD *de* CIENCIAS
de la COMPUTACIÓN

Maestría en Ciencias De la Computación

AGRUPAMIENTO DE OBJETOS BASÁNDOSE EN SU
APARIENCIA

Tesis que para obtener el grado de Maestro en Ciencias de la Computación
presenta:

OSCAR ANTONIO DE LEÓN GÓMEZ

Directores de tesis:

Dr. Luis Carlos Altamirano Robles

Dr. Mario Mauricio Bustillo Díaz

Octubre de 2012

A Esther.

RESUMEN

En este trabajo de tesis se propone una novedosa metodología para el agrupamiento de modelos basado en su apariencia. Esta forma de agrupamiento permite que, de acuerdo a un valor prefijado para la cantidad de objetos en las clases, se pueda construir una jerarquización de los mismos, de tal forma que el resultado de la ejecución del algoritmos con diferentes parámetros, genera un árbol de agrupamiento que al ser analizado revela una similitud interesante con la forma en que el humano agrupa objetos sin supervisión alguna. Aunque se muestra una serie de ejemplos de agrupamientos, junto con un análisis de los mismos, más trabajo es requerido en esta dirección para lograr construir un sistema de agrupamiento automático, similar al que posee el ser humano.

No hay que entender esta compulsión a construir conceptos, especies, formas, fines, leyes (un mundo de casos idénticos) como si nos permitiera ordenar el mundo real, sino como la compulsión de organizar un mundo para nosotros mismos en la que nuestra existencia se hace posible: de ese modo creamos un mundo que es calculable, comprensible, simple, etc, sólo para nosotros

Federico Nietzsche

AGRADECIMIENTOS

A mis asesores: el Dr. Luis Carlos Altamirano Robles y el Dr. Mario Mauricio Bustillo Díaz.

A mi comité tutorial: Los doctores: Guillermo De Ita Luna, Abraham Sánchez López y Fernando Zacarías Flores.

A mis compañeros de las áreas de computación matemática y computación aplicada a las ciencias y la ingeniería. En especial a mis amigos Alberto Román Flores y José Luis Serapio Palacios.

ÍNDICE GENERAL

1	INTRODUCCIÓN	1
1.1	Apariencia de un objeto	2
1.2	Modelos basados en apariencia	3
1.3	Sistema de adquisición de modelos	5
2	PRELIMINARES MATEMÁTICOS	7
2.1	Creación de modelos utilizando la transformada de Hotelling	7
2.2	Medidas de similaridad	9
2.2.1	Distancia euclidea	11
2.2.2	Distancia de Fréchet	11
2.2.3	Distancia Manifold-Manifold	12
2.2.4	Distancia geodesica	14
2.3	Parche lineal maximal	15
2.3.1	Medida de distancia entre ejemplares	16
2.4	Cálculo de ángulos principales	17
2.4.1	Definición	17
2.5	Distancia por variaciones	18
2.6	Algoritmos de agrupamiento	18
2.6.1	Algoritmo K-means	18
2.7	Redes Neuronales	20
2.8	Agrupamiento espectral	20
2.8.1	Notación	21
2.8.2	Diferentes tipos de grafos de similaridad	21
2.8.3	Matrices asociadas a un grafo	22

2.8.4	Algoritmos de agrupamiento espectral	23
3	PROBLEMÁTICA A RESOLVER	25
3.1	Problemas comunes en el paradigma basado en apariencia	25
4	PROPUESTA DE SOLUCIÓN	30
4.1	Algoritmo Manifold K-means	30
4.2	Otras propuestas de solución.	33
5	PRUEBAS	35
5.1	Resultados obtenidos	36
5.1.1	Clustering espectral utilizando la distancia entre Manifolds(MMD)	36
5.1.2	K-means para centroides de manifolds	38
5.1.3	Clustering espectral para centroides de manifolds	40
6	CONCLUSIONES Y TRABAJO FUTURO	43
A	APENDICE	46
A.1	Redes neuronales	46
A.1.1	El perceptrón	47
A.1.2	Aprendizaje con redes neuronales	47
A.1.3	El perceptrón multicapa	48

ÍNDICE DE FIGURAS

Figura 1	Izquierda: imagen de un objeto; derecha: gráfica de curvas de nivel <i>manifolds</i> para el mismo objeto utilizando los valores de los tres primeros eigenvectores y dos grados de libertad: θ_1 y θ_2 [Nay96].	3
Figura 2	ejemplos de manifolds extraídos de COIL-20	9
Figura 3	Ejemplos de bases de modelos	26
Figura 4	Secuencias de imágenes utilizadas en aplicaciones de vigilancia y seguimiento	27
Figura 5	Ejemplos de modelos de apariencia de gestos de la mano	27
Figura 6	Agrupamiento de Manifolds utilizando Manifold K-means	32
Figura 7	Jerarquización de un conjunto de datos	34
Figura 8	Interfaz gráfica del sistema	35
Figura 9	Indices de los objetos de COIL-20 como se muestran en los resultados	36
Figura 10	Jerarquización de un conjunto de datos utilizando MMD como medida de similaridad	37
Figura 11	Uno de los grupos resultantes utilizando MMD	38
Figura 12	Jerarquización de un conjunto de datos utilizando kmeans sobre los centroides de los manifolds	39
Figura 13	Uno de los grupos resultantes utilizando kmeans sobre los centroides	40

Figura 14	Jerarquización de un conjunto de datos	41
Figura 15	Uno de los grupos resultantes	41
Figura 16	Jerarquización de un conjunto de datos con $v = (3, 2, 3)$	42
Figura 17	Ejemplos de agrupamientos no previstos	44

INTRODUCCIÓN

La visión es un fenómeno sensorial que se produce en el ojo como respuesta a los estímulos luminosos del ambiente. La luz incide en la retina donde células especializadas, llamadas bastones y conos, la transforman en impulsos eléctricos que son transmitidos al cerebro por medio del nervio óptico. Esta señal es procesada luego por la corteza visual. Ya en los últimos años de 1700, Emmanuel Kant estableció que la percepción en general es una representación de la realidad objetiva externa, donde la vista tiene un papel preponderante. Así, lo que experimentamos como *ver* es sólo una parte de la percepción humana, una parte de un modelo del mundo real captado por el sistema *cerebro-ojo*, esto último ha sido confirmado empíricamente por la psicología cognitiva y las neurociencias. En nuestra vida cotidiana es fácil pasar por alto que estamos continuamente bombardeados por cantidades enormes de estímulos visuales. Para hacer frente a esta avalancha de información, el cerebro las organiza en experiencias visuales que son almacenadas en la memoria y se encuentran asociadas con otras funciones como la emoción y el lenguaje. De esta manera, el cerebro puede encargarse de la tarea de clasificar e identificar los objetos que percibe visualmente.

En los inicios del reconocimiento de objetos mediante imágenes digitales, se utilizó ampliamente el paradigma geométrico, dada la amplia disponibilidad de algoritmos para la detección de bordes y esquinas. A pesar de arrojar buenos resultados, se pone de manifiesto que utilizar únicamente la forma

para reconocer objetos tiene sus limitantes, por ejemplo, si tenemos dos cubos iguales en forma y tamaño pero uno es rojo y el otro azul. Obviamente cuando se le presenten estas instancias a un sistema de reconocimiento que tome sólo en cuenta las características geométricas, éste nos dirá que ambos son el mismo objeto pues tienen la misma forma.

Esto sucede porque se pierde mucha información inherente al objeto. En general ambos métodos tienen sus ventajas y sus desventajas pero parecen complementarse. Así, tener un sistema clasificador de objetos más natural requiere de:

- Maneras adecuadas de diferenciar entre objetos modelados a través de conjuntos de imágenes digitales (medidas de similaridad/disimilaridad).
- Algoritmos de agrupamiento capaces de poner modelos de objetos en clases en lugar de clasificar sólo puntos como los algoritmos usuales.

1.1 APARIENCIA DE UN OBJETO

La apariencia de un objeto está dada por la combinación de su forma, sus propiedades de reflectancia, su posición ante la cámara en la escena y la iluminación presente en ésta [Nay96]. Es importante notar que mientras las dos primeras son inherentes al objeto, la posición del elemento frente al sensor y la iluminación pueden variar de una escena a otra [Nay96]. Así mismo, es claro que esta definición implica que la apariencia de un objeto en un momento determinado, (o al menos, de parte de ese objeto), puede deducirse a partir de una imagen digital en la que se capture ese elemento en una escena específica, (con una luz determinada y un cierto ángulo de la cámara). De igual manera, podrá emplearse entonces un conjunto de imágenes para generar un modelo del objeto en diversas situaciones y poses frente a la cámara.

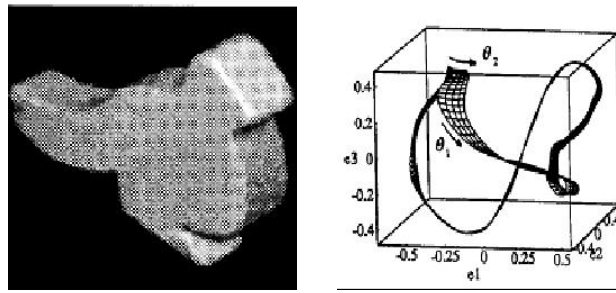


Figura 1: Izquierda: imagen de un objeto; derecha: gráfica de curvas de nivel *manifolds* para el mismo objeto utilizando los valores de los tres primeros eigenvectores y dos grados de libertad: θ_1 y θ_2 [Nay96].

En la figura se puede ver la representación gráfica de la apariencia del objeto en el espacio de apariencia para la vista que contiene esta imagen digital.

Formalmente podemos definir la apariencia de un objeto como una función $\sigma : [0, 2\pi] \rightarrow R^{nm}$, parametrizada por el ángulo de rotación del objeto frente a la cámara. Esta función es llamada también la trayectoria definida por la apariencia del objeto en el espacio de matrices $R_{n \times m}$. Cuando la trayectoria es parametrizada por más de una variable, el azimut de la cámara y el ángulo de rotación, la apariencia de un objeto se define formalmente como una superficie parametrizada por D , donde D es algún dominio en $R^{n_1 \times n_2 \times \dots \times n_k}$, También es llamada *manifold*. Así, la apariencia de un objeto puede ser tratada de manera abstracta como un objeto matemático y es estudiada por la topología.

1.2 MODELOS BASADOS EN APARIENCIA

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 (como el contorno del objeto) 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.

Debido a estos inconvenientes, se propuso a mediados de los años 90 del siglo pasado, una representación de objetos basándose en su apariencia, que preserva además de las características geométricas, la textura y el color del objeto.

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. 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.).

Aunque en este enfoque se tienen algunos problemas con estos modelos, se ha demostrado en algunos estudios tanto en humanos como en animales, como por ejemplo el estudio citado en [Nay96] realizado en macacos, que los sistemas biológicos trabajan con representaciones basadas en vistas. De aquí que se es razonable reproducir esta forma de visión en sistemas automáticos.

1.3 SISTEMA DE ADQUISICIÓN DE MODELOS

Para la adquisición de modelos de objetos se requiere obtener un conjunto de imágenes digitales (vistas) de cada objeto en particular para luego obtener de ellas el modelo correspondiente a ese conjunto de objetos o *base de modelos*.

El sistema consta básicamente de [Reyo6]:

- Un tornamesa controlado por un motor a pasos
- Una fuente de iluminación constante
- Una cámara digital

Además, este sistema debe situarse en un entorno que asegura completa obscuridad, con el fin de mantener constantes las características de iluminación, ya que los modelos de apariencia son muy sensibles a los cambios de éstas. Es necesario introducir algunas condiciones de ideales para la captura de estos modelos, a saber:

1. Cada imagen contendrá sólo un objeto
2. Todas las imágenes tendrán la misma resolución y profundidad en bits.
3. El objeto a modelar se presentará libre de oclusiones.

4. las imágenes están normalizadas en escala y brillantez[Robo2]

Se ha determinado empíricamente que para tener un modelo *completo* sólo se requieren de 36 a 72 imágenes del objeto. A partir de este modelo se puede obtener un modelo no uniforme[Robo2] que mantiene la información necesaria del objeto al mismo tiempo que se reduce el número de imágenes necesarias para la generación del modelo sin afectar de manera significativa el porcentaje de reconocimiento del sistema.

PRELIMINARES MATEMÁTICOS

2.1 CREACIÓN DE MODELOS UTILIZANDO LA TRANSFORMADA DE HOTELLING

Dada la cantidad de imágenes que se utilizan para modelar un único objeto y a que en la práctica es común que se tengan grandes cantidades de objetos diferentes, se pone de manifiesto la necesidad de encontrar una representación comprimida de todas las imágenes necesarias. En este trabajo se utiliza una técnica frecuentemente utilizada en el reconocimiento de patrones: la transformada de Hotelling o, como es más conocida, Transformada PCA (Análisis de Componentes principales), por sus siglas en inglés [Nay96].

El procedimiento a seguir es el siguiente: [Nay96, Coo04]

1. Cada imagen se empila por columnas, dando como resultado un vector columna i_j de $m * n$ filas. Donde m y n son las dimensiones de la imagen original.
2. Se normaliza cada una de las imágenes obtenidas en el paso anterior dividiendo entre la norma correspondiente:

$$\hat{i}_j = \frac{i_j}{\|i_j\|} \quad (1)$$

3. Se obtiene el vector promedio de todas las imágenes

$$i_m = \frac{\sum_{j=1}^r \hat{i}_j}{r} \quad (2)$$

4. Se resta el promedio a cada imagen
5. Se obtiene la matriz P, donde cada columna es uno de los vectores antes calculados. P es de dimensiones (m*n) filas por r columnas
6. Se obtiene la matriz de covarianza implícita, que es de dimensiones r*r, mucho menor a la matriz de covarianza:

$$\bar{Q} = P^T * P \quad (3)$$

7. Se obtienen los eigenvalores y eigenvectores de $\bar{Q}, \lambda_1, \dots, \lambda_k$, de los cuales se obtienen los correspondientes a la matriz de covarianza, mediante:

$$\lambda_i = \bar{\lambda}_i \quad (4)$$

$$e_i = \bar{\lambda}_i^{-1/2} P \bar{e}_i \quad (5)$$

Después de este proceso se obtiene un conjunto de vectores propios o *eigenimágenes*, estos vectores y valores propios forman un *eigenespacio* de pocas dimensiones. Las dimensiones del eigenespacio corresponden al número de componentes principales que se desee utilizar. En la práctica son suficientes 20 como máximo. Se debe tomar en cuenta que este procedimiento debe incluir a todas las imágenes de todos los objetos. Al resolver el sistema de ecuaciones:

$$eA = P \quad (6)$$

Se obtiene la matriz A que contiene los coeficientes necesarios para reconstruir cualquier imagen de cualquier modelo. Este esquema de compresión muestra pérdidas de información insignificantes para el problema que nos atañe. En la figura 2 se muestra el resultado de la extracción de manifolds para dos objetos.

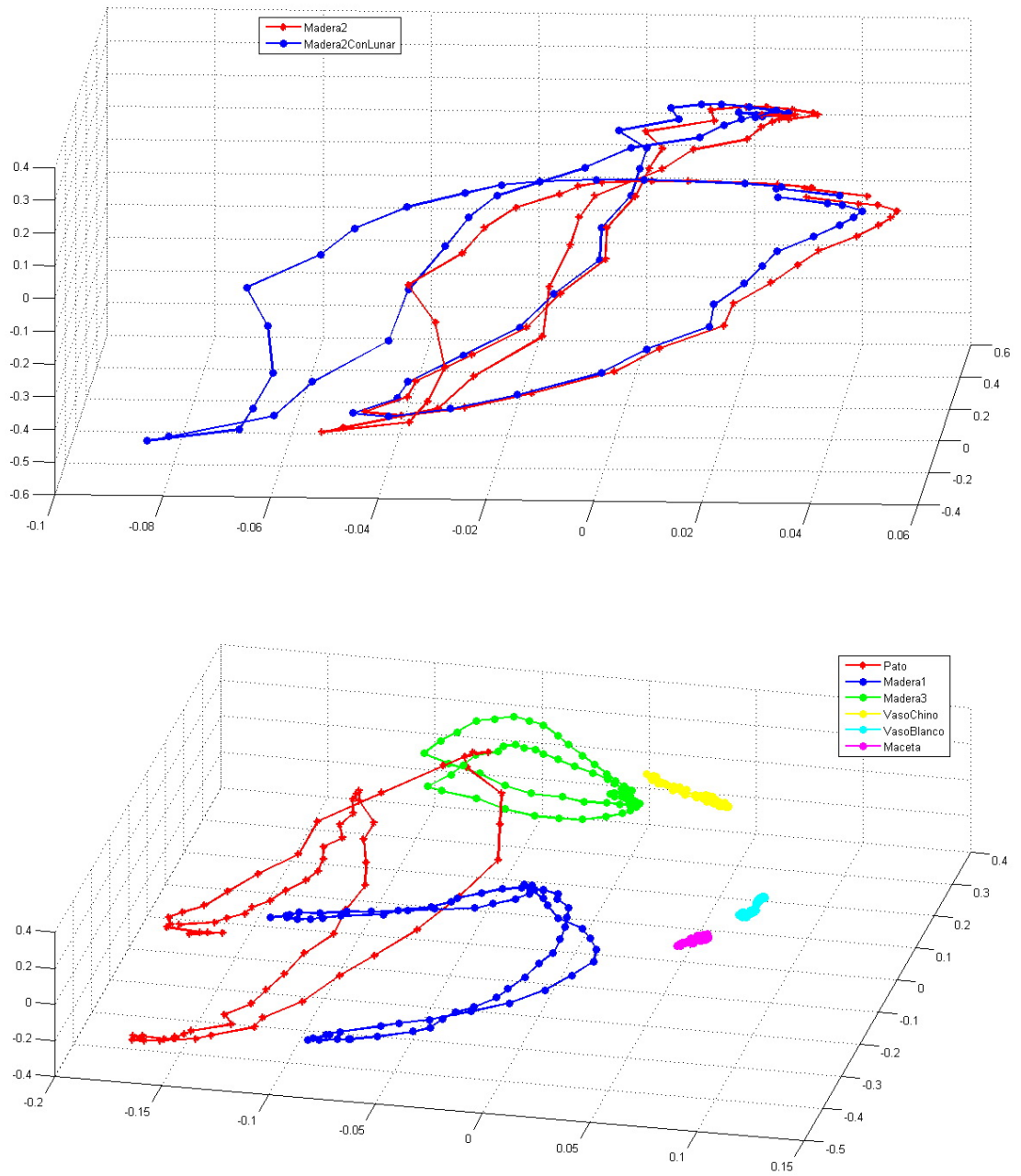


Figura 2: ejemplos de manifolds extraídos de COIL-20

2.2 MEDIDAS DE SIMILARIDAD

Como ya se dijo en la sección anterior, la matriz A contiene los *manifolds* de los objetos en la base de modelos. Estos definen una trayectoria parametrizada

en R^k donde k es el número de componentes principales seleccionados. El problema de agrupar modelos de objetos basados en apariencia debe tratar con esta representación compacta. Es indispensable encontrar la medida o conjunto de medidas de similaridad que sean necesarias para realizar una clasificación de objetos parecida a la que un ser humano haría. En la literatura lo común es encontrar medidas de similaridad entre objetos representados como un vector (columna o fila) con d características, almacenados en bases de datos de N objetos (dataset) [XW09]. La fuerza de la relación entre dos puntos de un *dataset* es indicada por un *coeficiente de similaridad*, entre más grande es este coeficiente, más relacionados están dichos puntos. Para que una función definida en los atributos de un objeto sea considerada una métrica debe cumplir las siguientes características:

1. Simetría
2. Reflexividad
3. Positividad
4. Cumplir con la desigualdad triangular.

Sólo cuando una función de similaridad no cumple con la desigualdad del triángulo, es llamada una semimétrica [XW09]. Existen muchas otras estructuras de similaridad sobre un dataset D [GG07] como puede ser la definida por un árbol o una partición de D . Estas medidas de similaridad son utilizadas extensivamente como parte de las técnicas corrientes de agrupamiento. Por mencionar las más importantes tenemos:

- La distancia euclídeana
- La distancia Manhattan
- La distancia de Mahalanobis

2.2.1 Distancia euclídeana

La distancia euclídeana es probablemente la medida de distancia más utilizada con datos numéricos. Si tenemos x y y , dos vectores en un espacio de dimensión d , la distancia euclídeana entre x y y se define como:

$$D(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (7)$$

Su uso en algoritmos de agrupamiento tiende a formar clusters hiperesféricos invariantes a translaciones y rotaciones en el espacio de características^o

2.2.2 Distancia de Fréchet

Esta distancia se usa para medir la similaridad de curvas en un espacio de dimensión d . Comúnmente cuando $d=2$ se aplica en matching de curvas parciales y en el reconocimiento de palabras en texto. La distancia de Fréchet está basada en la posición relativa de los puntos que forman una curva y está definida como sigue:

$$FD(f, g) = \inf_{\alpha, \beta} \max_{t \in [0,1]} d_p(f(\alpha(t)), g(\beta(t))) \quad (8)$$

donde f y g son las curvas que deseamos comparar. El uso de la posición relativa hace que sea sensible a las translaciones de ambas curvas. Para remediar esto se han introducido algunas mejoras como: la distancia de Fréchet dirigida o la isotópica.

Como medida de disimilaridad (entre más grande es el coeficiente, la diferencia entre las curvas es mayor) se tiene a la distancia de Hausdorff, que también es utilizada ampliamente para el reconocimiento de objetos que se define como sigue, Para dos curvas f y g :

$$HD(f, g) = \max_{t \in [0,1]} d_p(f(\alpha(t)), g(\beta(t))) \quad (9)$$

2.2.3 *Distancia Manifold-Manifold*

Como se puede ver, las medidas de similaridad anteriores sólo trabajan sobre vectores individuales de alguna dimensión d , lo que los hace difíciles de utilizar para el reconocimiento de objetos bajo el paradigma de apariencia. Por lo tanto, se hace necesario un nuevo enfoque para tratar con conjuntos de imágenes. La distancia *manifold-manifold* ha sido usada en el reconocimiento de rostros[RW08] donde, comparado con otros métodos tales como eigenfaces, Fisherfaces[PNBK97], vecinos más cercanos[HP04] o método de subespacio mutuo OY98], mejora enormemente el desempeño de la identificación de sujetos. En la literatura abundan los ejemplos de medidas de distancia entre puntos y entre subespacios pero no para manifolds. A continuación se enunciarán algunos conceptos necesarios para definir formalmente la distancia entre manifolds.

2.2.3.1 *Distancia punto a punto*

Denotada por $d(x_1, x_2)$ es tradicionalmente la distancia euclideana y otras que ya se han mencionado en apartados anteriores.

2.2.3.2 *Distancia punto a subespacio*

Se denota como $d(x_1, S)$, la distancia entre el punto x_1 y el subespacio S . También es llamada la distancia L2-Hausdorff y se define como:

$$d(x, S) = \min_{y \in S} \|x, y\| = \|x - x'\| \quad (10)$$

Donde x' es la proyección de x en S .

2.2.3.3 Distancia subespacio a subespacio

Se denota por $d(S_1, S_2)$ y en la actualidad no existe una definición unificada. Se ha utilizado el concepto de *ángulos principales* por su buen desempeño[BG73]. Con estos conceptos se definirán las siguientes medidas de distancia. La idea principal es que los manifolds no lineales están compuestos por subespacios lineales y que es la distancia de estos componentes la que define la similaridad entre dos manifolds. Formalmente se denotará como: $M = C_1, C_2, \dots, C_m$, donde M es un manifold compuesto de m subespacios lineales.

2.2.3.4 Distancia de un punto a un manifold

Se denota como $d(x, M)$ y se define de la siguiente manera

$$d(x, S) = \min_{C_i \in M} d(x, C_i) = \min_{C_i \in M} \min_{y \in C_i} \|x - y\| = \|x - x''\| \quad (11)$$

Donde x'' es llamada la proyección de x en el manifold.

2.2.3.5 Distancia de un subespacio a un manifold

Se denota como $d(S, M)$ y se define de la siguiente manera

$$d(S, M) = \min_{C_i \in M} d(S, C_i) \quad (12)$$

Lo que se reduce a encontrar la mínima distancia entre un subespacio y los elementos lineales del manifold.

2.2.3.6 Distancia de un manifold a un manifold

Se denota como $d(M_1, M_2)$ y se define de la siguiente manera

$$d(M_1, M_2) = \min_{C_i \in M_1} d(C_i, M_2) = \min_{C_i \in M_1} \min_{C'_j \in M_2} d(C_i, C'_j) \quad (13)$$

Claramente, la similaridad entre M_1 y M_2 es calculada como la similaridad de los modelos locales que mejor se adapten. En [RWo8] esta distancia se utiliza en el reconocimiento de rostros mediante conjuntos de imágenes. Aquí se utilizará para conseguir el agrupamiento de modelos de objetos basados en apariencia.

2.2.4 Distancia geodesica

Esta distancia está definida en teoría de grafos como el número de aristas en el camino mínimo entre dos vértices, donde si éstos no están conectados la distancia geodésica es convencionalmente definida como *infinita*[JBo3]. Para calcular la distancia geodésica primero debe construirse un grafo simétrico de vecindad conectando el punto(nodo) x_i con el punto x_j si éste es uno de los k -vecinos más cercanos del primero. Las aristas de este grafo están pesadas con la distancia euclidea entre los puntos vecinos. Luego, las distancias geodesicas son calculadas como la longitud del camino mínimo sobre las aristas utilizando alguno de los algoritmos disponibles como el algoritmo de Dijkstra o el de Floyd. Formalmente[XWo9], sea $D(x_i, x_j)$ la distancia euclidea entre x_i y x_j . La distancia geodésica $D_G(x_i, x_j)$ se inicializa:

$$D_G(x_i, x_j) = \begin{cases} D(x_i, x_j) & \text{si } x_i \text{ está conectado a } x_j \\ \infty & \text{en cualquier otro caso} \end{cases} \quad (14)$$

Luego, para cada punto x_k , donde $k = 1, 2, \dots, N$ y N es el número de puntos en el dataset, se calcula :

$$D_G(x_i, x_j) = \min(D_G(x_i, x_j), D_G(x_i, x_k) + D_G(x_k, x_j))$$

Lo que da como resultado la matriz de distancias geodésicas.

2.3 PARCHE LINEAL MAXIMAL

El concepto de parche lineal maximal(Maximal Linear Patch) o MLP está definido como un conjunto de puntos que definen un subespacio lineal maximal donde la perturbación lineal de estos puntos está dada naturalmente como la desviación entre la distancia euclidiana y la distancia geodésica. La idea es partir el conjunto de datos en varios MLPs que contengan conjuntos de imágenes muy parecidas entre si. Estos parches se calculan partiendo de un punto aleatorio del dataset hasta que la condición de linealidad se pierda. Este procedimiento garantiza la linealidad local de los modelos y controla adaptativamente el número de modelos calculados. Formalmente[RWo8], sea $X = x_1, x_2, \dots, x_N$ un conjunto de datos, en nuestro caso son puntos que conforman manifolds, donde $x_i \in R^D$ y N es el número de puntos en X . Se busca partir X en una colección disjunta de MLPs C_i tales que:

$$X = \bigcup_{i=1}^m C_i, \quad (15)$$

$$C_i \cup C_j = \emptyset (i \neq j, i, j = 1, 2, \dots, m) \quad (16)$$

$$C_i = x_1^{(i)}, x_2^{(i)}, \dots, x_{N_i}^{(i)} \left(\sum_{i=1}^m N_i = N \right) \quad (17)$$

donde m es el número de parches y N_i es el número de puntos en el parche. Para realizar el particionamiento es necesario calcular la matriz de distancias euclidianas D_E y la matriz de distancias geodésicas D_G . El algoritmo utilizado en el cálculo se muestra a continuación.

Algoritmo 1 Construcción de MLPs

```

 $i = 1, C_i = \emptyset, X_T = \emptyset, X_T = \emptyset, X_R = X$ 
while  $X_T \neq \emptyset$  do
  Seleccionar aleatoriamente un punto en  $X_R$  como semilla  $x_1^{(i)}$ ,
   $C_i = \{x_1^{(i)}\}, X_R = X_R - \{x_1^{(i)}\}$ 
  for  $\forall x_m^i \in C_i$  do
    Identificar cada uno de los  $k$  vecinos más cercanos  $x_c$  como candidato
    if  $x_c \in X_R (D_G(x_c, x_n^{(i)})/D_E(x_c, x_n^{(i)})) < \theta$  then
       $C_i = C_i \cup \{x_c\}$ 
       $X_R = X_R - \{x_c\}$ 
    end if
  end for
   $X_T = \bigcup_{j=1}^i C_j, X_R = X - X_T; i \leftarrow i + 1, C_i = \emptyset$ 
end while

```

La principal ventaja de este algoritmo es que sólo debe utilizarse una vez para cada manifold, con lo que se alcanza una gran eficiencia. Por otro lado, este método tiende a formar MLPs más grandes al principio y más pequeños al final lo que no parece afectar el rendimiento general del algoritmo[RWo8].

2.3.1 Medida de distancia entre ejemplares

Una vez obtenido el conjunto de los subespacios lineales de los objetos de la biblioteca, es necesario discriminar entre los modelos y un conjunto de imágenes de prueba para lograr el reconocimiento de objetos. Para este objetivo se utiliza a medida de similaridad entre ejemplares, que es definida como la correlación de dos ejemplares e_i y e_j mediante[RWo8]:

$$d_E(e_i, e_j) = \|e_i\| \|e_j\| / e_i^T e_j \quad (18)$$

Cuando se comparan dos conjuntos de imágenes, la distancia entre ejemplares nos muestra la similaridad entre la apariencia de los objetos.

2.4 CÁLCULO DE ÁNGULOS PRINCIPALES

Los ángulos principales tienen muchas aplicaciones importantes en estadística y análisis numérico. También se encuentra en la teoría de aproximación de mínimos cuadrados.

2.4.1 Definición

Sean F y G dos subespacios en el espacio unitario E^m , y supongamos que

$$p = \dim(F) \geq \dim(G) = q \geq 1 \quad (19)$$

El ángulo más pequeño $\theta_1(F, G) = \theta_1 \in [0, \pi/2]$ entre F y G es definido como:

$$\cos \theta_1 = \max_{u \in F} \max_{v \in G} u^H v, \quad \|u\|_2 = 1, \quad \|v\|_2 = 1$$

Supongamos que es máximo es alcanzado cuando $u = u_1$ y $v = v_1$. Entonces, $\theta_2(F, G)$ es definido como el ángulo más pequeño entre el complemento ortogonal de F con respecto a u_1 y de G respecto a v_1 . Continuando de esta manera hasta que uno de los subespacios quede vacío. Entonces tenemos que Los ángulos principales $\theta_k \in [0, \pi/2]$ entre F y G son calculados recursivamente para $k = 1, 2, \dots, q$ mediante

$$\cos \theta_k = \max_{u \in F} \max_{v \in G} u_k^H v_k, \quad \|u\|_2 = 1, \quad \|v\|_2 = 1 \quad (20)$$

sujeto a las restricciones

$$u_i^H = 0, \quad v_i^H = 0, \quad j = 1, 2, \dots, k-1$$

En [BG73] estos ángulos se calculan con un algoritmo numéricamente estable. En MATLAB se encuentra una función que nos permite realizar este cálculo

2.5 DISTANCIA POR VARIACIONES

Para dos subespacios C_i y C_j definimos una *medida de distancia por variaciones* como el promedio de sus correlaciones canónicas como sigue:

$$d_V(C_i, C_j) = r / \sum_{k=1}^r \sigma_k \quad (21)$$

donde $\sigma_1 \dots \sigma_r$ son las correlaciones canónicas y son obtenidos por el método reportado en [BG73] y r es el mínimo de los tamaños de los subespacios lineales.

2.6 ALGORITMOS DE AGRUPAMIENTO

2.6.1 Algoritmo K-means

El algoritmo K-means es uno de los más usados debido su sencillez y velocidad. La idea central es agrupar repetidamente los datos alrededor de los centroides calculados a partir de una asignación inicial de los datos en un número fijo k de grupos. La pertenencia a las clases cambia de acuerdo a la evaluación de una función de error hasta que este error no cambia significativamente o la pertenencia de los datos tampoco. La función de error se define como sigue[GG07]:

Sea D un conjunto de datos con n instancias, y sean C_1, C_2, \dots, C_k k cluster disjuntos de D . La función de error se calcula como:

$$E = \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu(C_i)) \quad (22)$$

Donde $\mu(C_i)$ es el centroide del cluster C_i y $d(.,.)$ es típicamente la distancia euclideana. A continuación se lista el algoritmo.

Algoritmo 2 Algoritmo K-means

Fase de inicialización

C_1, C_2, \dots, C_k es la partición inicial de D .

Fase iterativa

repeat

$d(i, j)$ = distancia entre objeto i y cluster j

$n_i = \arg \min_{1 \leq j \leq k} d_{ij}$

Asignar objeto i al cluster n_i

until Sin cambios en la pertenencia de los datos

Regresar los resultados.

Como una de las técnicas de agrupamiento más usadas, k-means tiene varias propiedades importantes[GG07]:

- Es eficiente para conjuntos de datos grandes
- Termina frecuentemente en un óptimo local
- Es capaz de identificar clusters convexos(hiperesféricos) y linealmente separables.
- Sólo se puede utilizar para datos numéricos
- Depende en gran medida de la selección de los centroides iniciales.

El algoritmo no funciona bien para datos de alta dimensionalidad[XW09]. También conocer el número de clusters es un problema ya que se asume que el usuario ya conoce éste parámetro y no existe un método universal y eficiente que nos ayude a seleccionar k . Se han propuesto distintas maneras de inicializar los centroides iniciales tales como kmeans genético que utiliza un paso k-means como operador de cruce[XW09, KNM99]. Por otro lado, k-means es susceptible al ruido. Esto es problemático ya que estos puntos modifican las formas de los clusters y también se han propuesto modificaciones para solventar estos inconvenientes tales como k medioides[ECY00]. Estos medioides forman parte del conjunto de datos original y que tiene la mínima distancia promedio a todos los demás puntos en el cluster al que representan.

2.7 REDES NEURONALES

Se probaron las redes neuronales como otra opción para lograr la clasificación de los objetos de la base de modelos **COIL-20**. Sin embargo, dado que es necesario entrenar la red antes de pedirle que clasifique los datos y que los ejemplos de cada clase deben ser escogidos a mano y esto hizo imposible el objetivo inicial de que el sistema fuera totalmente automático. Por lo tanto se descartó esta técnica a pesar de que para la clasificación inicial dió resultados aceptables. Una revisión de la teoría de las redes neuronales se encuentra en el apéndice A de este trabajo.

2.8 AGRUPAMIENTO ESPECTRAL

La teoría de grafos puede ser usada para realizar particiones de datos. En estas técnicas se utiliza una medida de similaridad entre los datos para formar

una matriz de adyacencias con peso. Intuitivamente, lo que se quiere es encontrar una partición en el grafo donde las aristas de los nodos que se encuentran en diferentes grupos tengan bajo peso y que tengan pesos altos cuando se encuentren en el mismo grupo[Luxo7]. Para formalizar esta idea, introduciremos algunos formalismos.

2.8.1 Notación

Sea $G = (V, E)$ un grafo no dirigido con un conjunto de vértices $V = v_1, \dots, v_n$. Se asume que cada arista tiene asociado un peso positivo. La matrix de adyacencias pesadas $W = w_{ij}, i, j = 1, \dots, n$, donde si $w_{ij} = 0$ los vértices i y j no están conectados. Dado que G es no dirigido se tiene que $w_{ij} = w_{ji}$. Además, el *grado de un vértice* está definido como:

$$d_i = \sum_{j=1}^n w_{ij} \quad (23)$$

La matriz diagonal que tiene los grados de los vértices es llamada *matriz de grado* y se denota como D .

2.8.2 Diferentes tipos de grafos de similaridad

Cuando se construye un grafo de similaridad la meta es modelar las relaciones de vecindad local entre los datos. Existen varias maneras de construir la matriz con similaridades o distancias mutuas:

Grafo de ϵ -vecindad: Todos los pesos de aristas por encima o por debajo de cierto umbral ϵ se igualan a cero suponiendo que los puntos cercanos se encuentran a cierta distancia(cuando menos ϵ).

Grafo de k -vecinos más cercanos: En este caso se conservan en el grafo las

aristas que corresponden a los k vecinos más cercanos de cada punto. Sin embargo, esto nos lleva a tener un grafo dirigido puesto que la relación de vecindad no es simétrica. Se puede arreglar este problema incluyendo a v_j en los vecinos más cercanos de v_i si éste último es vecino de v_j para así tener un *grafo de k vecindades mutuas*.

Grafo completamente conectado Aquí simplemente se conectan todos los vértices entre sí. Esta construcción es útil cuando la medida de similaridad nos dice algo acerca de la naturaleza de la relación de vecindad local.

Hasta el momento no existe ningún trabajo hasta ahora sobre la prevalencia de una construcción sobre otra [Lux07, Ude11]

2.8.3 Matrices asociadas a un grafo

Ya definimos la matriz de adyacencias pesadas y la matriz de grados, ahora definiremos la matriz laplaciana no normalizada de un grafo como:

$$L = D - W \tag{24}$$

Esta matriz tiene varias propiedades interesantes. Es simétrica y positiva semidefinida y el número de eigen vectores de esta matriz está directamente relacionado con la cantidad de componentes conectadas del grafo [Lux07]

2.8.3.1 Matrices laplacianas normalizadas

En la literatura existen dos matrices laplacianas normalizadas. Cada una está estrechamente relacionada con la otra y se definen como:

$$L_{sym} = D^{1/2} L D^{1/2} \qquad L_{rw} = D^{-1} L \tag{25}$$

La matriz L_{sym} se denota así porque es simétrica y la matriz L_{rw} está muy relacionada a los recorridos aleatorios.

Como en el caso de la matriz no normalizada, la multiplicidad del primer eigenvalor de la matriz laplaciana está relacionada con el número de componentes conectadas.

2.8.4 Algoritmos de agrupamiento espectral

Estos algoritmos están íntimamente relacionados con los cortes normalizados[BJo3], Random Walks[MSo1] y también es posible expresar el algoritmo k-means kernelizado como un problema de agrupamiento espectral[Dhio4]. A continuación se muestran los algoritmos más comunes.

2.8.4.1 Agrupamiento espectral no normalizado

Asumiremos que nuestro conjunto de datos está compuesto de N puntos, x_1, \dots, x_n , los cuales pueden ser arbitrarios. Medimos la similaridad entre todos los pares de puntos mediante alguna función de similaridad, la cual debe ser simétrica y positiva, y construimos la matriz de similaridad S .

Algoritmo 3 Agrupamiento espectral no normalizado

Entrada: $S \in R^{n \times n}$, k .

Construir el grafo de similaridad

Calcular la matriz laplaciana L

Calcular los primeros k eigenvectores u_1, \dots, u_k de L .

Sea $U \in R^{n \times k}$, la matriz que contiene los vectores u_1, \dots, u_k como columnas.

Tomando cada renglón de U como un objeto, agrupar con el algoritmo k-means en k clusters

Salida: Los clusters C_1, \dots, C_k

El siguiente algoritmo utiliza la matriz L_{sym} . Como puede verse se utiliza un paso de normalización por filas que no está presente en el algoritmo anterior.

2.8.4.2 Agrupamiento espectral normalizado

Algoritmo 4 Agrupamiento espectral normalizado

Entrada: $S \in R^{n \times n}$, k .

Construir el grafo de similaridad

Calcular la matriz laplaciana L

Calcular los primeros k eigenvectores u_1, \dots, u_k de L .

Sea $U \in R^{n \times k}$, la matriz que contiene los vectores u_1, \dots, u_k como columnas.

Tomando cada renglón de U como un objeto, agrupar con el algoritmo k -means en k clusters

Salida: Los clusters C_1, \dots, C_k

Estos algoritmos se ven muy similares siendo sólo diferentes en la matriz laplaciana que es utilizada. En ambos el truco principal es cambiar la representación de los puntos x_i a los puntos $y_i \in R^k$. Es por las propiedades de las matrices laplacianas que este cambio es útil. Una vez que la representación ha sido cambiada, el algoritmo k -means no tiene problemas al identificar el número de clusters solicitado.

PROBLEMÁTICA A RESOLVER

3.1 PROBLEMAS COMUNES EN EL PARADIGMA BASADO EN APARIENCIA

Las técnicas basadas en apariencia implican un muestreo uniforme y discreto de imágenes que no toman en cuenta la complejidad del objeto, así que el número de imágenes necesarias para la representación de un objeto se supone como una constante independiente[Robo2] en el caso de escenas con parámetros controlados. Por otro lado, en la práctica se tiene que las condiciones en las que se capturan los modelos no son ideales y cambian constantemente. Esto provoca que las tareas de identificación en situaciones reales se enfrenten a tres problemas principales:

- El espacio de búsqueda de imágenes para la estimación de pose es grande.
- El espacio de almacenamiento en disco es prohibitivo.
- Los manifolds resultantes de la proyección a un espacio de menor dimensionalidad tienden a traslaparse.

En condiciones ideales, un modelo de apariencia *completo* requiere, según[ea99], de seis grados de libertad para considerar las variantes de la posición del objeto y dos para considerar la dirección de la iluminación. Este modelo utiliza imágenes de $128 * 128$ píxeles que da como resultado vectores de dimensión 16384. Además del problema de almacenamiento de los modelos, es necesario

tratar el problema del reconocimiento de objetos en una imagen recién presentada al sistema. La primera aproximación al reconocimiento es el comparar cada imagen de los modelos almacenados con la nueva instancia calculando la diferencia entre las imágenes, lo que es obviamente ineficiente. También se puede utilizar otras medidas como la suma de diferencias cuadradas[Ber09] lo que no resuelve los problemas antes mencionados.

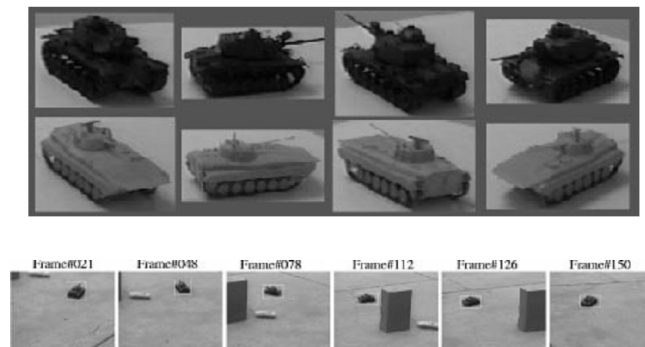


Figura 3: Ejemplos de bases de modelos

Uno de los problemas prácticos encontrados en los cuales se utiliza el paradigma de apariencia es, por ejemplo, el uso de estas técnicas en grandes bases de datos de imágenes infrarrojas para aplicaciones militares de identificación de vehículos en misiones de reconocimiento. Estas bases de datos contienen hasta 17000 imágenes pertenecientes a 10 clases diferentes donde los objetos aparecen en gran cantidad de posiciones frente al sensor, lo que provoca traslapes en los manifolds después de aplicar la transformada de Hotelling, lo que hace el reconocimiento más difícil[BLo1]. En casos como estos se tiene que recurrir a los métodos geométricos que tienen el grave inconveniente de depender del punto de vista en el que se toman las imágenes.

Este tipo de problemas se presentan también en los sistemas de vigilancia y seguimiento donde es necesario identificar un objeto como perteneciente a un conjunto de hipótesis o identificarlo en una secuencia que contiene varios otros

objetos, en este caso se tiene que utilizar la apariencia del objeto en conjunto con un esquema geométrico.

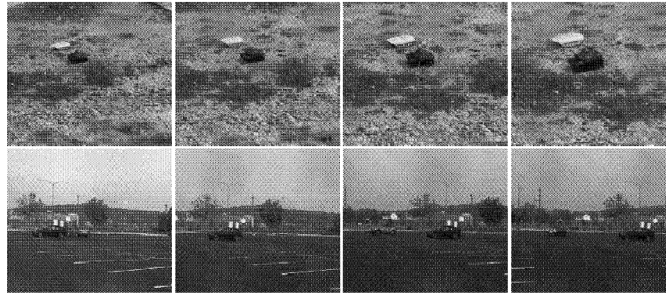


Figura 4: Secuencias de imágenes utilizadas en aplicaciones de vigilancia y seguimiento

Otra aplicación práctica es el reconocimiento de gestos y signos de la mano para traducción del lenguaje de signos para sordomudos o en la computación ubicua [JMo1] donde los problemas antes mencionados de espacio de almacenamiento, traslape de manifolds y tamaño del espacio de búsqueda aparecen como graves desventajas.

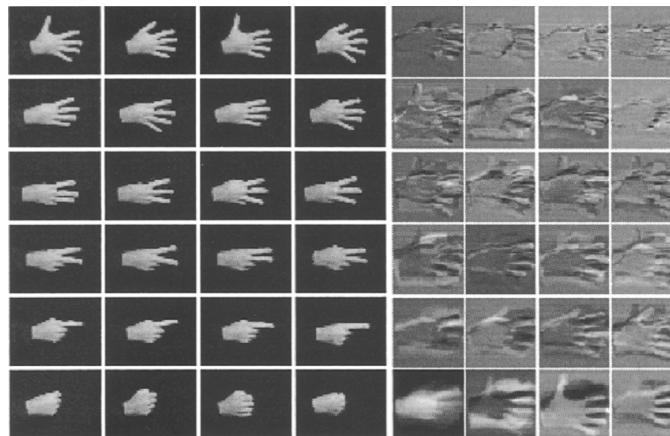


Figura 5: Ejemplos de modelos de apariencia de gestos de la mano

Para lograr el reconocimiento de objetos en cualquier posible circunstancia es necesario contar con modelos desde todos los puntos de vista posibles. Aunque esto último es imposible, se pueden obtener modelos de los objetos en

condiciones que se consideren típicas para el problema que se quiere resolver incluyendo situaciones en las cuales el objeto de interés puede estar ocluido parcialmente o puede aparecer en compañía de otros como en la figura 3.

Sin embargo los problema de espacio de almacenamiento, traslape de manifolds y del tamaño del espacio de búsqueda permanecen. Este problema se ha intentado resolver mediante diversas técnicas como el muestreo no uniforme [Rob02] y la representación paramétrica de modelos [Nay96] o FisherFaces [PNBK97].

Dados estos problemas se pone de manifiesto la necesidad de contar con una manera de reducir el espacio de búsqueda, reducir el número de imágenes necesarias y resolver el problema del traslape de manifolds. En este trabajo de tesis nuestro objetivo es desarrollar un procedimiento capaz de crear una jerarquía en un conjunto de modelos basados en apariencia que le permita al usuario definir y escoger un conjunto de clusters que le ayuden en la aplicación que desee. Esta jerarquía puede servir como instrumento para lograr un reconocimiento e identificación más rápidos ya que sería posible presentarle al sistema un modelo nuevo e indicarle que lo agregue en la rama y nivel correctos de la jerarquía o decidir si está o no dentro de los datos considerados. Para lograr estos objetivos se deben tomar en cuenta varios aspectos tales como la mejor manera de partir el conjunto de datos y encontrar una forma efectiva de comparar modelos de apariencia entre sí.

Clasificar un conjunto de objetos mediante sus respectivos modelos de apariencia en grupos donde una característica o conjunto de características los haga muy similares entre si y diferentes entre todos los demás y repetir este procedimiento con cada grupo obtenido. Estas particiones sucesivas tienen el

objetivo de encontrar una estructura de tipo jerárquico.

Existen dos tipos de algoritmos jerárquicos: divisivos y aglomerativos. Los algoritmos aglomerativos comienzan cuando cada objeto se encuentra en su propio grupo. Entonces se mezclan sucesivamente los grupos más parecidos hasta que todos los objetos pertenecen a un sólo grupo. Para realizar esta mezcla se utiliza la distancia mínima, la distancia máxima, la distancia promedio o la suma del error cuadrático, este último se usa en el método de Ward[XW09]. Algunas de las ventajas de estos métodos son: la flexibilidad, se puede utilizar cualquier medida de distancia o similaridad y su versatilidad. No obstante tienen sus desventajas tales como en el caso de los algoritmos divisivos donde las comparaciones necesarias para encontrar la mejor partición hacen de la mayoría de éstos demasiado complejos computacionalmente. Existen otros intentos como el clustering bayesiano[HGo5] sin embargo todos estos métodos nunca han sido usados para agrupar modelos de apariencia, por lo que se desarrollará una estrategia divisiva para agrupar conjuntos completos de puntos pertenecientes a un objeto.

PROPUESTA DE SOLUCIÓN

Para lograr el objetivo de este trabajo de tesis, que es obtener una estructura jerárquica en una base de modelos de apariencia de objetos, se proponen varios métodos con la intención de utilizarlos de manera iterativa, primero sobre el conjunto original para conseguir una primera partición y luego, hacerlo para cada una de las clases obtenidas en el paso anterior hasta que cada un sólo objeto forme su propia clase o se alcance el número de niveles requeridos.

Primeramente se modificó el conocido método de agrupamiento *K-means* [XW09, GG07] para que en lugar de sólo trabajar con puntos aislados, fuera capaz de agrupar *manifolds* tratandolos no como puntos aislados sino como una entidad completa. Luego se exploró la conveniencia de transformar los *manifolds* representaciones concretas y utilizar algoritmos conocidos y por último usando la medida de similaridad entre manifolds como entrada para los algoritmos de agrupamiento espectral.

4.1 ALGORITMO MANIFOLD K-MEANS

La modificación al algoritmo *K-means* consistió principalmente en utilizar la distancia Manifold-Manifold en lugar de la distancia euclidiana para determinar la similaridad de los objetos. En la figura 6 se muestra un esquema que ilustra una primera versión del resultado esperado de esta propuesta. El

algoritmo propuesto se lista a continuación:

Algoritmo 5 Algoritmo Manifold K-means

Conjunto de datos D , número de clusters k , dimensión d , criterio de agrupamiento, número de niveles de la jerarquía, h

Fase de inicialización

C_1, C_2, \dots, C_k es la partición inicial de D .

Fase iterativa

repeat

$d(i, j)$ = distancia **Manifold-Manifold** entre Manifold i y cluster j

$n_i = \arg \min_{1 \leq j \leq k} d_{ij}$

Asignar objeto i al cluster n_i

until Sin cambios en la pertenencia de los datos

for $i \leftarrow 1 \rightarrow h$ **do**

repeat

$d(i, j)$ = distancia entre cluster i y Supercluster j

$n_i = \arg \min_{1 \leq j \leq k} d_{ij}$

Asignar cluster i al supercluster n_i

until Sin cambios en la pertenencia de los datos

end for

Regresar los resultados.

Se trata de un algoritmo que utiliza k-means con una medida de distancia diferente a la euclídeana y que además utiliza la partición inicial realizada en

la primera pasada para agrupar los clusters resultantes en *superclusters* tantas veces como niveles de jerarquía sean solicitados. Nótese además que se hace necesario agregar un parámetro de entrada que corresponde al numero de grupos a obtener en cada nivel

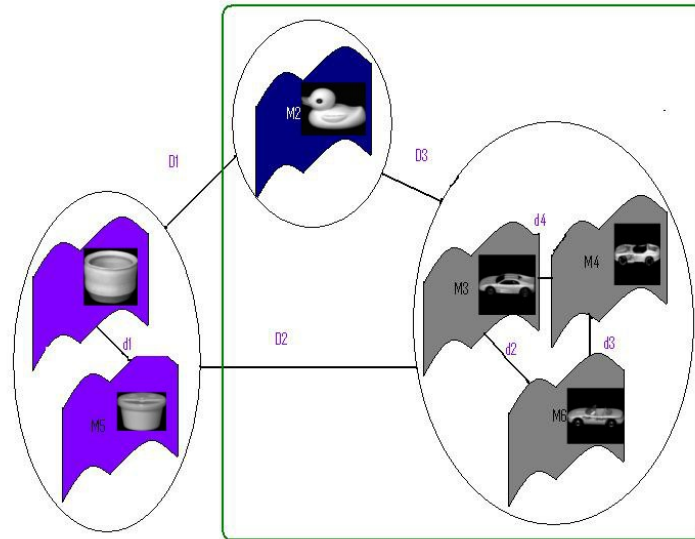


Figura 6: Agrupamiento de Manifolds utilizando Manifold K-means

También se probará el desempeño de otros algoritmos de agrupamiento que pueden ser utilizados como un sustituto del algoritmo k-means para el particionamiento de los datos. Entre los algoritmos que servirán para este efecto son: Agrupamiento espectral con distancia entre manifolds. Otra idea que se tomará en cuenta es la de calcular el promedio de cada modelo para obtener el centroide de una esfera de radio σ que es la desviación estándar de tal manifold. El objetivo de esta transformación del manifold es tratarlo como un todo encerrado en una esfera y compararlo con los demás puntos utilizando métodos conocidos de agrupamiento.

4.2 OTRAS PROPUESTAS DE SOLUCIÓN.

Para probar otros algoritmos se tomará como base el algoritmo presentado en la sección anterior. Ahora, además de los parámetros especificados se agregará el algoritmo que se desea probar y el número de grupos a encontrar para cada grupo en cada nivel de la jerarquía.

Algoritmo 6 Algoritmo Jerarquizar Base de modelos

Conjunto de datos D , número de niveles de la jerarquía, h , grupos a encontrar para cada grupo en el nivel, V , Algoritmo a utilizar, alg

Realizar la Primera partición: $C = C_{11}, C_{12}, \dots, C_{1V(1)}$ de D

Fase iterativa

repeat

repeat

$C_{j,i} = \text{algoritmo}(D, V(\text{nivel}))$, dividir cada grupo del nivel en el número especificado de grupos

 Asignar los Grupos obtenidos al nivel de la jerarquía

until Todos los grupos del nivel hayan sido particionados

until Los niveles requeridos han sido obtenidos

Regresar los grupos obtenidos para cada nivel.

Ahora podemos obtener cualquier número de niveles de jerarquización y cada grupo obtenido en el paso anterior será dividido en los grupos indicados para formar una estructura como la que se muestra a continuación:

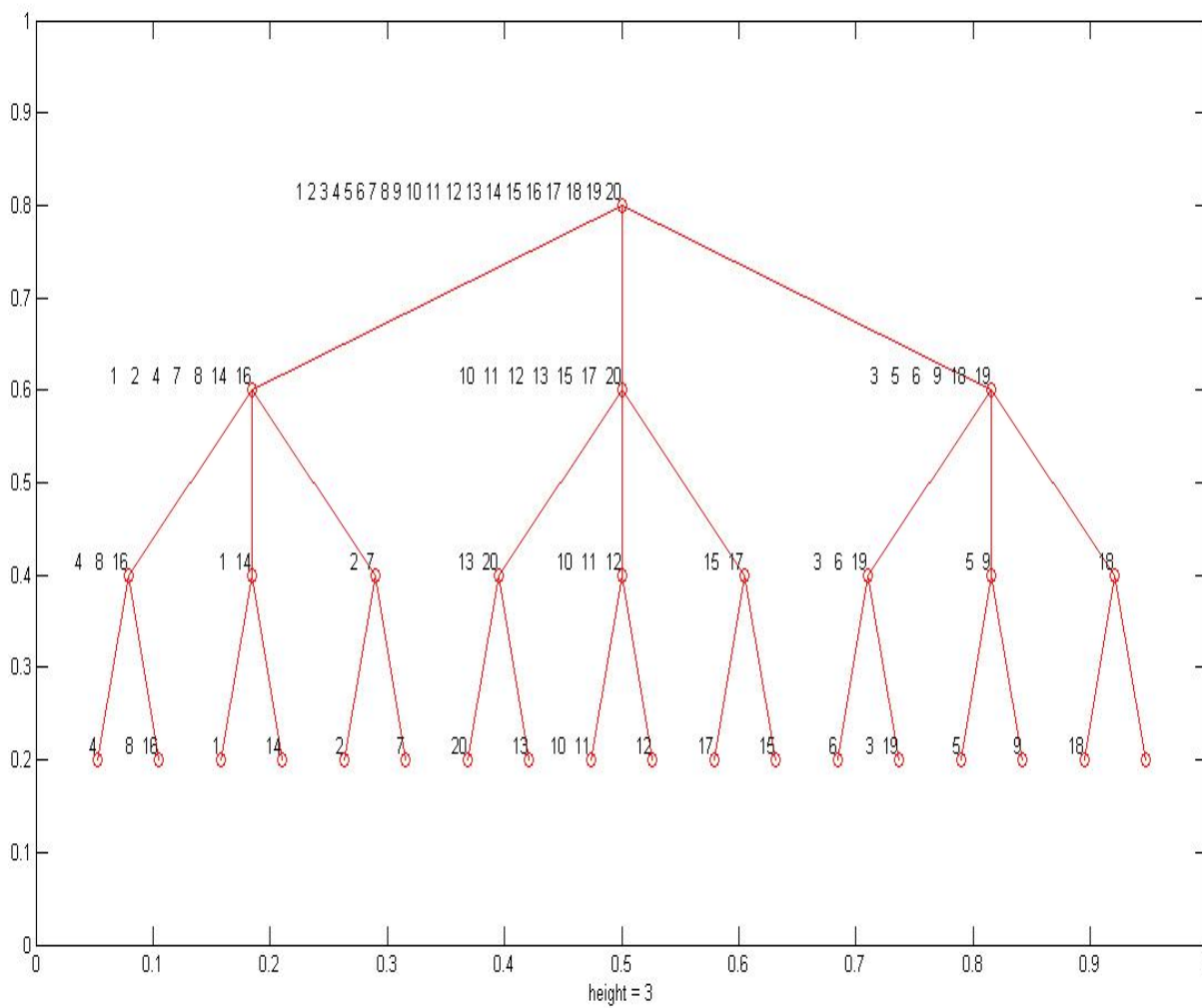


Figura 7: Jerarquización de un conjunto de datos

PRUEBAS

El objetivo de este trabajo de tesis es obtener una clasificación de objetos tal como la que una persona haría. Para conseguir que esta clasificación sea automática se construyó un sistema básico para introducir diferentes parámetros al algoritmo propuesto. Las pruebas fueron realizadas sobre el clásico conjunto de modelos de apariencia suministrados por la Universidad de Columbia, **COIL-20**. Los resultados obtenidos fueron validados comparandolos con los obtenidos por un usuario de manera visual, esto con la finalidad de comprobar que la clasificación obtenida es similar o no.

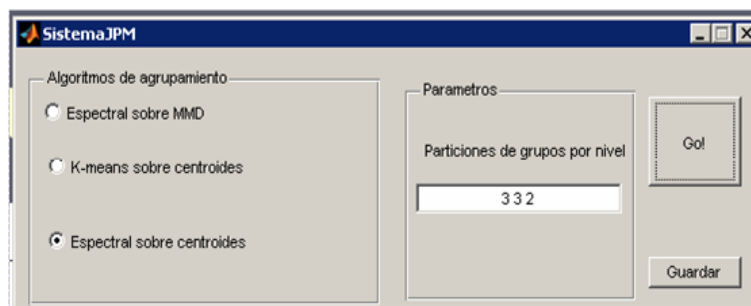


Figura 8: Interfaz gráfica del sistema

Como puede verse, el sistema sólo permite seleccionar el algoritmo a utilizar para la clasificación y un vector de enteros. El número de niveles de la jerarquía es calculado como el tamaño del vector, que contiene el número de grupos en que se va a dividir cada grupo en cada nivel. Estos valores son aplicados de izquierda a derecha. El resultado es mostrado luego en forma de árbol, donde cada nodo está etiquetado con el número correspondiente al objeto

según la siguiente figura. El sistema fue ejecutado varias veces para cada

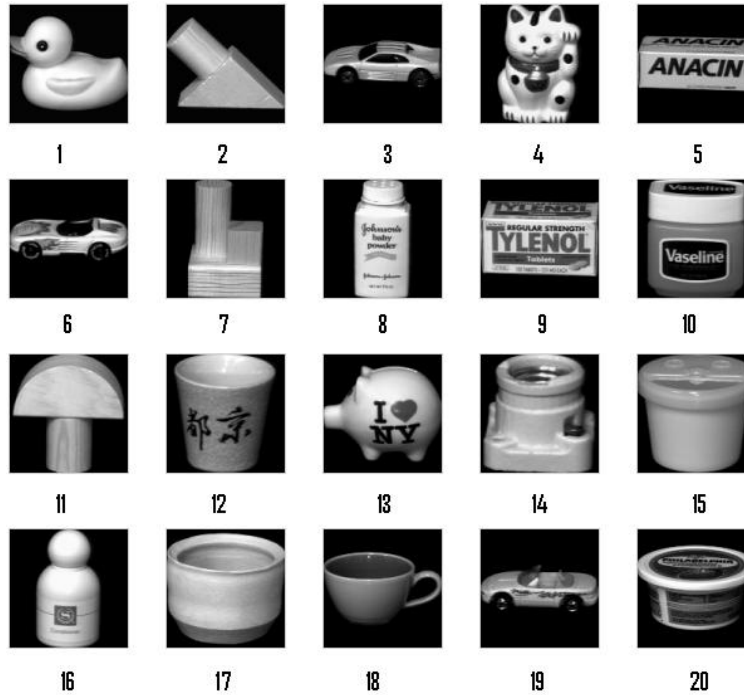


Figura 9: Indices de los objetos de COIL-20 como se muestran en los resultados

uno de los algoritmos y con diferentes números de niveles y particiones por nivel, encontrándose, de manera experimental, que en el caso particular de la biblioteca utilizada, tres niveles y no más de tres particiones por grupo son suficientes para lograr una buena clasificación.

5.1 RESULTADOS OBTENIDOS

5.1.1 Clustering espectral utilizando la distancia entre Manifolds(MMD)

La primera propuesta fue utilizar esta medida de similaridad dentro del algoritmo k-means lo cual no arrojó los resultados deseados debido a las limitaciones inherentes a $\tilde{\chi}^2$. Así que para probar la efectividad de la medida de similaridad, también se probó usando la matriz de distancias

entre los objetos como entrada para el algoritmo de clustering espectral. Los resultados obtenidos no son tan satisfactorios como los que se logran cuando los centroides de los manifolds son utilizados para la representación. Aún así, los resultados son muy similares en algunos casos.

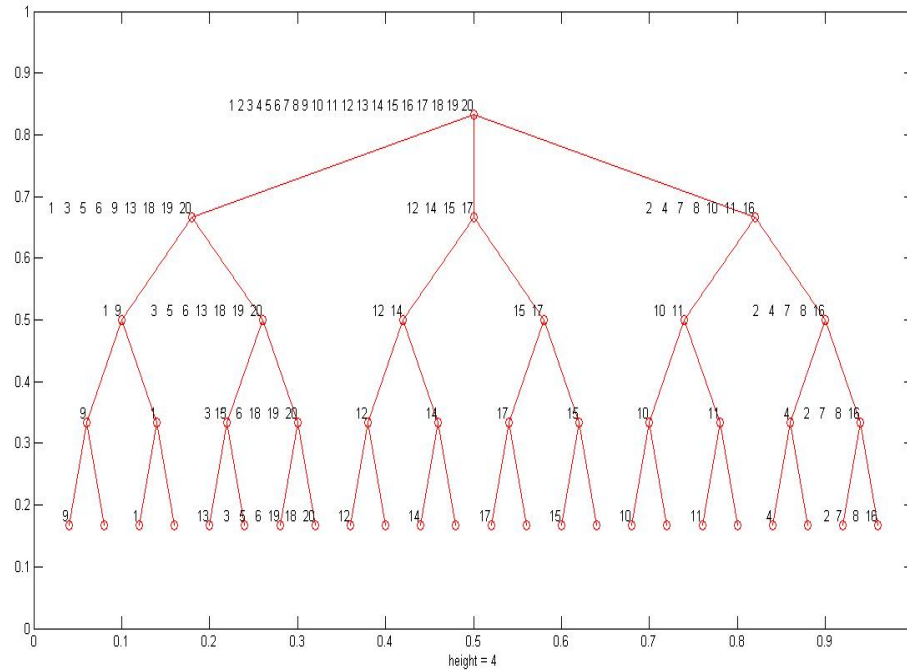


Figura 10: Jerarquización de un conjunto de datos utilizando MMD como medida de similaridad

En la figura siguiente se puede apreciar uno de los grupos calculados por el algoritmo. Note que la clasificación de los objetos 4,8 y 16 como pertenecientes a una misma clase no cambiará aplicando los demás métodos propuestos, sólo cambiarán los objetos con los que se agrupan en niveles superiores de la jerarquía esto es debido a que el cálculo de la distancia entre manifolds le da un peso mayor a la apariencia de los objetos así que aquellos en los cuales el tono de gris dominante es muy parecido son agrupados en la misma clase como queda de manifiesto en el caso de las piezas de madera. También esta medida es capaz de reflejar algunas características especiales de los objetos como la

forma que en el caso de la figura la mayoría comparte (los objetos son más altos que anchos, casi del mismo gris y tienen casi la misma clase de simetría) lo que puede ser un indicio de que los resultados pueden mejorarse mezclando de alguna manera estos resultados a aquellos obtenidos utilizando sólo los bordes.

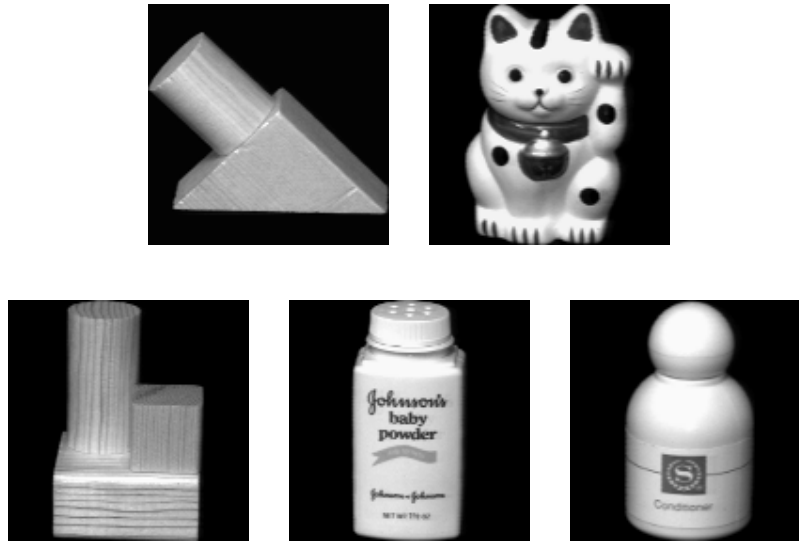


Figura 11: Uno de los grupos resultantes utilizando MMD

5.1.2 *K-means para centroides de manifolds*

Dado que *k-means* es uno de los métodos de agrupamiento más utilizados y estudiados de los últimos tiempos, se tomó en cuenta para el desarrollo de este trabajo de tesis, ya no modificado como era la intención inicialmente sino aplicándolo sobre un conjunto de datos transformados a una representación por centroides. Si los resultados de la sección anterior son regulares, los que se obtienen en este caso son mejores con los mismos niveles de jerarquía y más grupos por nivel. Seguidamente podemos ver un ejemplo de los resultados obtenidos.

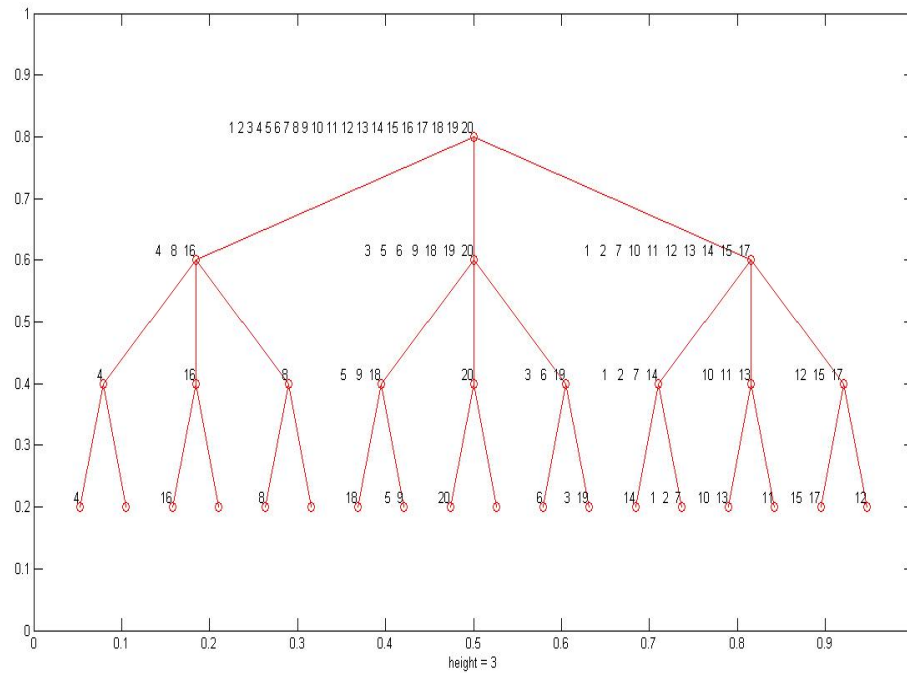


Figura 12: Jerarquización de un conjunto de datos utilizando kmeans sobre los centroides de los manifolds

De nuevo algunos de los grupos calculados con los otros algoritmos se repiten con la inclusión de diferentes objetos. Esto es debido a que al calcular los centroides de los manifold se pierde información acerca de la apariencia y la forma del objeto y la distancia entre puntos adquiere mayor relevancia. Esta es la principal desventaja de utilizar los centroides de los modelos para realizar la clasificación. No obstante existe alguna similitud basada en el tono de gris entre los objetos que conforman el grupo representado en la siguiente figura.

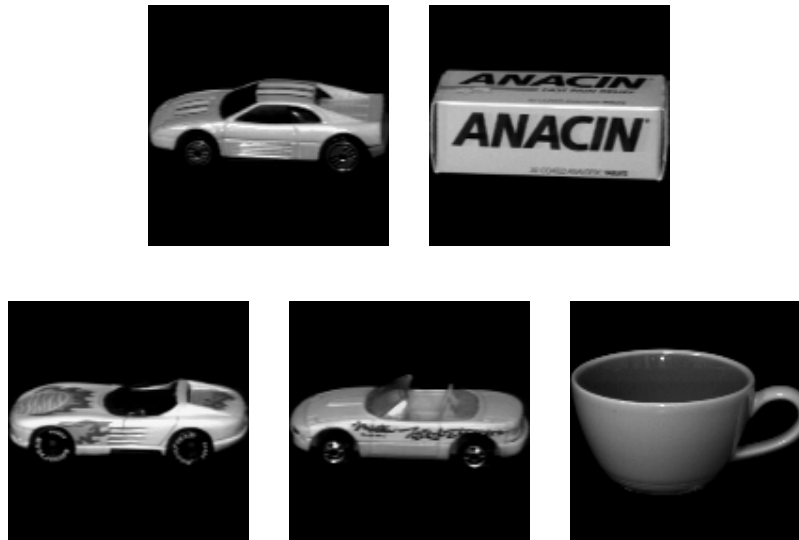


Figura 13: Uno de los grupos resultantes utilizando kmeans sobre los centroides

Realizar un cambio de representación de los modelos calculando los centroides nos permite utilizar sin modificaciones los algoritmos existentes pero al mismo tiempo conserva todos los inconvenientes innatos de estos mismos. Para evitar en lo particular los derivados de utilizar k-means, utilizaremos el agrupamiento espectral sobre los centroides.

5.1.3 *Clustering espectral para centroides de manifolds*

A continuación se muestran algunos de los resultados obtenidos en las pruebas del algoritmo. Para esta prueba en particular se usaron tres niveles y un vector $v = (3, 3, 2)$

Destacan en esta clasificación los grupos de extrema derecha y extrema izquierda compuestos por los objetos 1, 2, 4, 7, 8, 14, 16 y 35691819 respectivamente. Estos grupos están formados por objetos muy similares entre sí y al mismo tiempo presentan diferencias como en el caso del grupo formado por los objetos siguientes:

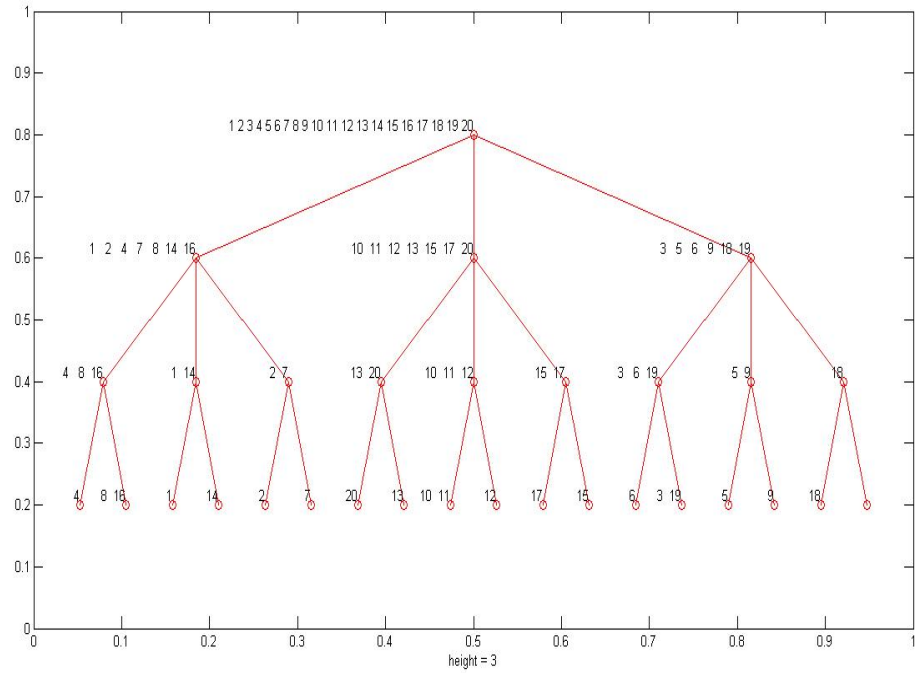


Figura 14: Jerarquización de un conjunto de datos



Figura 15: Uno de los grupos resultantes

Todos los objetos de este grupo son parecidos entre sí por la forma o numéricamente por el tono de gris dominante. Por esto último, el algoritmo

agrupó la taza (objeto 18) junto a los autos (objetos 3 6 y 19) y las cajas de medicamento(objetos 5 y 9). En el siguiente nivel de la jerarquía podemos observar que los autos, las cajas y la taza han sido separadas en su propia clase dejando a esta última alejada del resto.

Lo mismo sucede cuando se utiliza el vector $v = (3,2,3)$ para realizar las particiones. La única diferencia es que la última partición deja a casi todos los objetos es su propia clase. Sin embargo, los resultados son prácticamente los mismos como se puede ver en la siguiente figura. Los resultados son muy

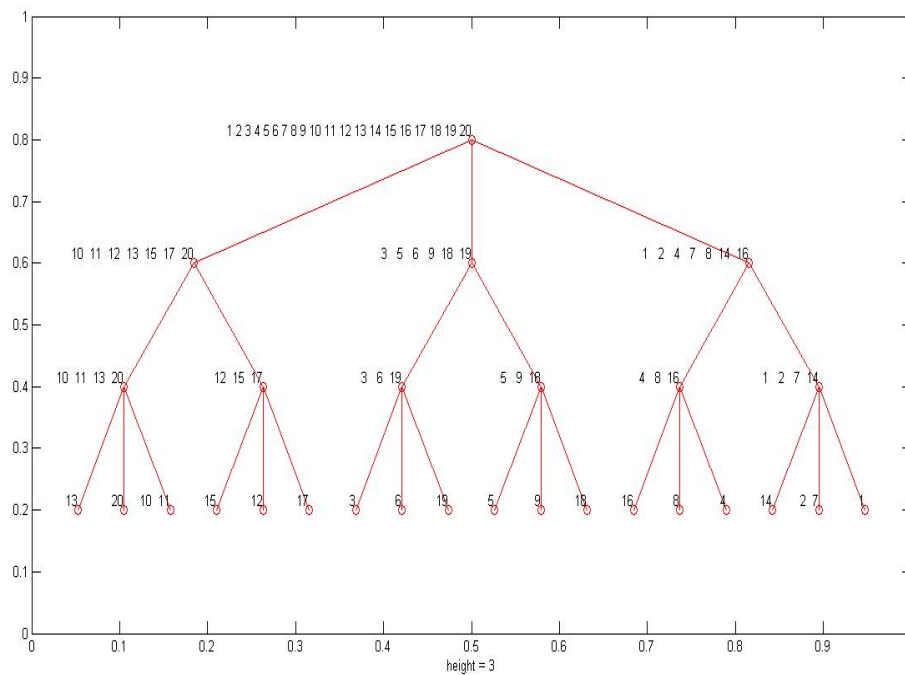


Figura 16: Jerarquización de un conjunto de datos con $v = (3,2,3)$

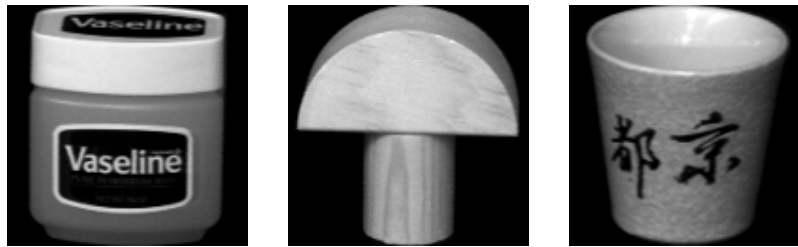
similares a los obtenidos en la sección anterior y es debido a que en realidad los algoritmos utilizados están muy relacionados por lo que es de esperarse que algunos de los grupos calculados por cada algoritmo se compartan.

CONCLUSIONES Y TRABAJO FUTURO

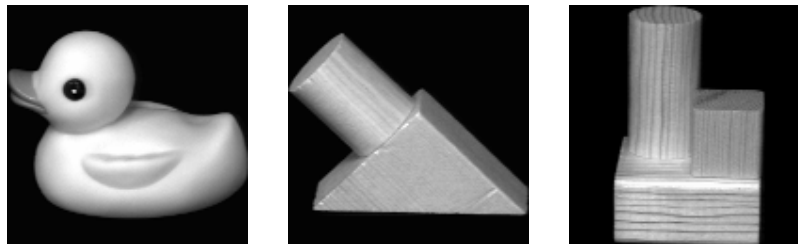
Se ha presentado una propuesta para realizar el agrupamiento y la jerarquización de un conjunto de objetos tridimensionales, que pueden ser identificados tanto individualmente como por el grupo al que pertenecen, de acuerdo a las características presentes en su apariencia; esta forma de agrupamiento trata de asemejarse a la forma en que el ser humano agrupa un conjunto de objetos, cuando no cuenta con mayor información que la visual, obtenida por simple inspección del conjunto de objeto, esto es, cuando no hay supervisión alguna. Hasta donde se sabe, es quizá la primera vez que esta clase de problema dentro del paradigma de apariencia es abordado, dada la complejidad del problema. Se espera que esta propuesta sirva de base para futuras propuestas en el campo del reconocimiento de objetos en particular, y de la visión artificial en general, además de que múltiples aplicaciones pueden ser derivadas de esta propuesta, tantas como el humano requiere en su vida diaria cuando clasifica y jerarquiza a: sus compañeros de trabajo, de escuela, de su familia, etc., así como objetos de su vida cotidiana: los camiones del servicio público que abordará, la comida que comerá, el auto que comprará, etc.

Aunque el algoritmo es del tipo divisivo, su complejidad es polinomial puesto que los algoritmos utilizados (kmeans y espectral) nos aseguran que los grupos formados son muy parecidos entre sí y muy diferentes a los que conforman otros grupos con lo cual se evita comprobar combinatoriamente cual es la mejor partición. No obstante, cuando un objeto ha sido clasificado

erróneamente, es imposible asignarlo a otra clase y en el mejor de los casos puede ser segregado del grupo al calcularse el nivel siguiente. Por otro lado, el uso de una medida de similitud entre objetos arrojó resultados muy parecidos a los obtenidos por los otros métodos ya que la biblioteca de modelos así lo permitía. Además se pudo comprobar que el sistema es capaz de igualar los resultados que obtiene el usuario y además es capaz de detectar similitudes que no son evidentes a simple vista. En la figura siguiente se muestran algunos de estos resultados.



(c) Objetos numéricamente similares



(f) Objetos similares en forma y apariencia

Figura 17: Ejemplos de agrupamientos no previstos

Este trabajo de tesis ha tenido como objetivo desde su comienzo, la exploración de propuestas para realizar clasificación de objetos bajo el paradigma basado en apariencia. Los resultados obtenidos han mostrado la utilidad de medidas de similitud alternativas a las clásicas y la transformación de los datos como medio para obtener mejores resultados. Sin embargo, queda mucho trabajo por hacer en este campo en el cual este trabajo pretende ser pionero. Algunos aspectos a explorar en trabajos futuros son:

- **Utilizar modelos basados en apariencia únicamente para los bordes de los objetos.** Con el objetivo de utilizarlo como método complementario
- **Seleccionar diferentes algoritmos para aplicarlos a cada nivel.** Es posible que los resultados mejoren si se usan diferentes algoritmos en cada nivel de la jerarquía
- **Probar con otras bibliotecas de modelos y comparar los resultados.** Es imprescindible probar con diferentes bibliotecas para delimitar el alcance de la propuesta desarrollada en este trabajo y modificarla para hacerla aplicable en problemas más generales.
- **Analizar automáticamente los resultados obtenidos.** Para comparar cuantitativamente las clasificaciones obtenidas por el sistema con las realizadas por un ser humano
- **Mejorar la representación de la estructura jerárquica** Explorar la posibilidad de calcular el dendrograma del árbol.
- Investigar los posibles campos de aplicación de esta propuesta.

Todos estos puntos pueden ser tomados como base para trabajos posteriores con el fin de encontrar aplicaciones, detectar problemas y generar otras propuestas que sirvan para extender el conocimiento que actualmente se tiene en las áreas de visión computacional y algoritmos de agrupamiento.

APENDICE

A.1 REDES NEURONALES

La evolución ha dotado al cerebro humano de características altamente deseables que no están presentes en las computadoras actuales, ni siquiera en aquellas con gran cantidad de procesadores paralelos. Algunas de estas características pueden ser:

- Paralelismo masivo
- Habilidad de aprender
- Habilidad de generalización
- Adaptividad
- Procesamiento inherente de información conceptual
- tolerancia a fallos

Estas propiedades han llevado al desarrollo de entes computacionales inspirados en las redes neuronales biológicas. Se han realizado numerosos esfuerzos para desarrollar programas inteligentes basados en la arquitectura Von Neumann que no han logrado reproducir una inteligencia artificial de propósito general [JMM96]. Como sistemas masivamente paralelos altamente interconectados, las redes neuronales artificiales intentan reproducir los principios

estructurales del cerebro humano para lograr imitar los resultados que éste puede ofrecer.

A.1.1 El perceptrón

En 1943 McCulloch y Pitts propusieron un modelo computacional para una neurona artificial[MP43]. Esta neurona matemática calcula una suma pesada de las n señales de entrada y genera una salida que es 1 si la suma es superior a cierto umbral u . De otra manera, la salida es 0. Matemáticamente,

$$y = \theta\left(\sum_{j=1}^n w_j x_j - u\right) \quad (26)$$

Donde θ es una función de paso y w_j es el peso de la sinapsis asociado a la j -ésima entrada. Debido a que el modelo de McCulloch y Pitts hace varias simplificaciones, el perceptrón sólo es capaz de clasificar grupos que son linealmente separables[TKo6]. Un problema que no puede resolverse usando un perceptrón es el *problema XOR*, para ello se introdujo el perceptrón multicapa.

A.1.2 Aprendizaje con redes neuronales

El algoritmo de aprendizaje de una red neuronal es el procedimiento mediante el cual ciertas reglas de aprendizaje se utilizan para actualizar los pesos de las sinápsis. Dependiendo de la aplicación de la red neuronal, ésta es capaz de realizar tres formas de aprendizaje: supervisado, no supervisado e híbrido entre las dos anteriores. En el aprendizaje supervisado, se le provee a la red de la salida deseada para cada patrón de entrada. Durante el aprendizaje la salida deseada y puede no ser igual a la salida deseada d así que para corregir este

error se utiliza la señal de error $(d - y)$ para modificar los pesos sinápticos y reducir el error gradualmente.

Un perceptrón con una neurona con pesos ajustables $w_j, j = 1, \dots, n$ y dado el vector de entrada $x = (x_1, \dots, x_n)$, la entrada de la neurona es:

$$v = \sum_{j=1}^n w_j x_j - u \quad (27)$$

La salida y del perceptrón es 1 si $v > 0$ y 0 en otro caso. Así en un problema de clasificación en dos clases, el perceptrón asigna un patrón a una clase si $y = 1$ y a la otra cuando $y = 0$. La ecuación:

$$\sum_{j=1}^n w_j x_j - u = 0 \quad (28)$$

Define la frontera de decisión entre las dos clases. Rosenblatt desarrolló un procedimiento de aprendizaje para determinar los pesos sinápticos y los umbrales dado un conjunto de datos. El algoritmo es el siguiente:

Como ya se dijo anteriormente, un perceptrón de una sola capa sólo puede clasificar conjuntos de datos linealmente separables. En la práctica nos interesa trabajar con conjuntos de datos que no son necesariamente linealmente separables.

A.1.3 El perceptrón multicapa

Una neurona es capaz de trazar una línea recta o un hiperplano entre dos clases linealmente separables. En el caso específico del problema XOR, no es posible encontrar una separación lineal y por ello surge la idea de tener varias neuronas y así trazar varias fronteras de decisión entre los datos. Las neuronas se organizan en varias capas, la capa de entrada, varias capas ocultas y la capa de salida. Esta arquitectura recibe el nombre de *perceptrón multicapa* o *redes feed-forward* y es capaz de realizar separaciones de complejidad arbitraria.

Con el desarrollo de el algoritmo de retropropagación(back-propagation), los perceptrones multicapa se han vuelto muy populares. El algoritmo se muestra a continuación:

Algoritmo 7 Algoritmo de retropropagación

Require: Inicializar aleatoriamente los pesos de la red

repeat

for all ejemplo e en el conjunto de entrenamiento **do**

 Calcular la respuesta de la red

 Calcular el error para cada salida de la red

 Regresar el error desde la salida a la capa oculta

 Ajustar los pesos de la capa oculta

 Ajustar los pesos de las entradas de la red

end for

until Todos los ejemplos clasificados o se alcanzó el criterio de paro

return Regresa la red ya entrenada

BIBLIOGRAFÍA

- [Ber09] Barrientos Flores Berenice. Inferencia en apariencia para el reconocimiento de objetos tridimensionales. Master's thesis, FCC-BUAP, 2009.
- [BG73] Å. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27:579594, 1973.
- [BJ03] Francis R. Bach and Michael I. Jordan. Learning spectral clustering. Technical Report UCB/CSD-03-1249, EECS Department, University of California, Berkeley, Jun 2003.
- [BL01] Rama Chellapa Baoxin Li. Model-based temporal object verification using video. *IEEE transactions on image processing*, 10:897–908, 2001.
- [Coo04] Taylor C.J Cootes, T.F. Statistical model of appearance for computer vision. Technical report, Imaging Science Biomedical Engineering, University of Manchester, March 2004.
- [Dhio4] Inderjit S. Dhillon. Kernel k-means, spectral clustering and normalized cuts. In -, pages 551–556. ACM Press, 2004.
- [ea99] Nayar et al. An experimental comparison of appearance and geometric based recognition. *NA*, NA:NA, 1999.
- [ECY00] Vladimir Estivill-Castro and Jianhua Yang. Fast and robust general purpose clustering algorithms. In *PRICAI*, pages 208–218, 2000.

- [GG07] Jianhong Wu Guojun Gan, Chaoqun Ma. *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability, SIAM, 2007.
- [HG05] Katherine A. Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *In Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005.
- [HP04] A. Hadid and M. Pietikäinen. From still image to video-based face recognition: An experimental analysis. *FG*, page 813818, 2004.
- [JB03] E. Guitter J. Boutier, P. Di Francesco. Geodesic distance in planar graphs. *Nuclear Physics B*, 663:535–567, 2003.
- [JM01] James L. Crowley Jerome Martin. An appearance based approach to gesture recognition. *GRAVIR-IMAG*, na:340–348, 2001.
- [JMM96] Anil K. Jain, Jianchang Mao, and K. Mohiuddin. Artificial neural networks: A tutorial. *IEEE Computer*, 29:31–44, 1996.
- [KNM99] K. Krishna and M. Narasimha Murty. Genetic k-means algorithm. *Trans. Sys. Man Cyber. Part B*, 29(3):433–439, June 1999.
- [Lux07] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.
- [MP43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [MS01] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *In Advances in Neural Information Processing Systems*, pages 873–879. MIT Press, 2001.

- [Nay96] Nene Nayar, Murase. Parametric appearance representation. *Nayar, S.K., Poggio, T., (Eds.) Early Visual Learning, Oxford University Press. New York, 1:131–160, 1996.*
- [OY98] K. Maeda O. Yamaguchi, K. Fukui. Face recognition using temporal image sequence. *FG*, page 318323, 1998.
- [PNBK97] J. P. Hespanha P. N. Belhumeur and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 19:711720, 1997.
- [Reyo6] Isaid Reyes. Sistema para la adquisición de imágenes, modelado y reconocimiento de objetos empleando apariencia. Tesis de Licenciatura, FCC-BUAP, 2006.
- [Robo2] Luis Carlos Altamirano Robles. *Metodologías para la reducción del número de imágenes requeridas para el reconocimiento de objetos articulados.* PhD thesis, IPN-CIC, 2002.
- [RWo8] Xilin Chen Wen Gao Ruiping Wang, Shiguang Shan. Manifold-manifold distance with application to face recognition based on image set. *IEEE*, nd:nd, 2008.
- [TKo6] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Third Edition.* Academic Press, Inc., Orlando, FL, USA, 2006.
- [Ude11] Madeleine Udell. Introduction to spectral graph theory, 2011.
- [XWo9] Rui Xu and Don Wunsch. *Clustering.* Wiley-IEEE Press, 2009.