

Face Recognition through a novel Indexing Method based on Permutations

Christian von Lücken¹ y Liz Jarmila González Brítez²
 Facultad Politécnica, Universidad Nacional de Asunción
 Asunción, Paraguay
 clucken@pol.una.py¹, liz.jarmila@gmail.com²

Abstract —In face recognition systems that work with databases of thousands of images it is not practical to compare the query image against each stored image to determine similarity. The increasing use of these systems in various fields creates the need to explore mechanisms for efficient and effective search in terms of use of computing resources and percentage of hits. In order to reduce the load of face recognition systems and improve their response time, several alternatives to reduce or eliminate the need for an exhaustive search of images were developed. Indexing methods are one of these alternatives. This paper presents a new indexing technique based on permutations which, optimizes storage of images and accelerates the search process to predict the similarity between objects. The proposed method shows improved behaviour when compared against other techniques representing the state of the art using the FERET database.

La fase de **clasificación** tiene el objetivo de asignar cada objeto a un conjunto de categorías predefinidas [8]. En esta fase se realiza la búsqueda por similitud cuya solución trivial consiste en el recorrido secuencial de la base de datos, comparando el objeto de búsqueda contra todos los objetos del conjunto y retornando aquellos objetos que se encuentren más cerca de éste. Con esta idea, los sistemas de reconocimiento facial que manejan bases de datos con miles y millones de imágenes se enfrentan al problema de ralentización del proceso de búsqueda ante la presencia de la **maldición de la dimensionalidad** [7]. Este problema ha convertido al estudio de la escalabilidad de los métodos de búsqueda en uno de los principales tópicos de investigación en la fase de clasificación de imágenes faciales.

I. INTRODUCCIÓN

Las investigaciones en el área de reconocimiento facial, desde sus inicios en 1960 [18], han abarcado varias disciplinas científicas y tecnológicas, además de haberse convertido en un tema de interés del público en general por su vasto campo de aplicación en la vida real.

Las fases que involucran un sistema de reconocimiento facial son pre-procesamiento, extracción de características y clasificación [15], [16], [17].

Durante el **pre-procesamiento** se realiza la normalización de las imágenes faciales. Los objetivos principales de esta etapa son: convertir las imágenes a dimensiones estándar, alinear la distribución de la luz en la imagen y determinar el ángulo de giro del rostro en la imagen. Los métodos que sirven para lograrlos son: la ecualización de histograma y el escalado de imagen.

En la **extracción de características** se realiza el mapeo de los objetos digitales en un espacio vectorial, donde cada coordenada representa el grado de presencia de una determinada característica en el objeto. Los objetivos principales de esta etapa son: la representación de la imagen facial y la reducción dimensional de la imagen mediante la extracción de las características principales. Algunos métodos que abordan dichos objetivos son: PCA (*Principal Component Analysis*), ICA (*Independent Component Analysis*), LDA (*Linear Discriminant Analysis*), EBGM (*Elastic Bunch Graph Matching*), entre otros [22], [23]. En el presente trabajo se utiliza el método PCA para extraer las características que contengan solo la información más relevante de las imágenes faciales, que luego son utilizadas en la fase de clasificación.

Los trabajos proporcionados por la comunidad científica, que abordan el problema de la escalabilidad de los métodos de búsqueda, plantean procesar previamente el conjunto de datos mediante la construcción de índices. Un índice es una abstracción de la base de datos que produce cierto tipo de orden de los objetos de búsqueda, optimizando el proceso de búsqueda y la recuperación eficiente de datos. Se han propuesto varios métodos de indexación de bases de datos de alta dimensión, entre los cuales se encuentran: estructuras de árboles, métodos de aproximación, métodos probabilísticos y hashing dependiente e independiente de los datos [2], [4].

Teniendo en cuenta que las métricas de distancias utilizadas originan un coste computacional muy elevado, los métodos de búsqueda mediante la indexación de la base de datos se esfuerzan en resolver las consultas reduciendo el número de evaluaciones de la función de distancia.

En el presente trabajo se propone un nuevo algoritmo para la etapa de clasificación del sistema de reconocimiento facial. El objetivo del algoritmo presentado es realizar una búsqueda y comparación de imágenes simplificada mediante la utilización de índices. El método propuesto en este trabajo se denomina **Emiap** (Espacios métricos indexados mediante algoritmos basados en permutación) y pertenece a la familia de los procedimientos que utilizan permutaciones para calcular índices. Para evaluar la propuesta, se la compara con otros métodos, considerando la tasa de aciertos y la cantidad de consultas utilizando el conjunto de imágenes FERET [5].

II. TRABAJOS RELACIONADOS A LA INDEXACIÓN DEL ESPACIO MÉTRICO

La indexación se realiza con el fin de reducir el costo de la búsqueda evitando comparaciones de distancia entre los elementos de la base de datos y la consulta. Las investigaciones realizadas en esta área se basan en la búsqueda de qué información almacenar para representar a cada elemento de la base de datos. Atendiendo a la recuperación de los datos, se tienen dos grandes grupos de algoritmos de búsqueda: exactos e inexactos.

Entre los algoritmos de búsqueda exactos están los métodos basados en pivotes que tienen un buen rendimiento en espacios de baja dimensión pero este se deteriora rápidamente conforme aumenta el tamaño del conjunto de datos [7] y los métodos basados en particiones que dividen el espacio en varias zonas, con el objetivo de que en el proceso de búsqueda se puedan descartar completamente algunas de estas zonas, sin necesitar para ello recurrir al cálculo de nuevas funciones de distancias.

Entre los algoritmos de búsqueda inexactos están los métodos aproximados y los métodos probabilísticos [1], [20]. Una variante de este tipo de métodos son los algoritmos basados en ordenamientos, en los cuales se propone un orden (por promisoriedad) en el que deben ser comparados los elementos de la base de datos [8].

Entre los métodos de búsqueda basados en ordenamientos están los algoritmos que pertenecen a la familia de algoritmos basados en permutaciones.

En [1] se presenta *Permutation Index (PI)* que tiene una alta precisión y reduce los cálculos de distancia entre el objeto buscado y los objetos candidatos que están en la base de datos. La idea de *PI* es identificar rápidamente los elementos prometedores y a medida que se degrade la calidad de la respuesta dejar de comparar elementos en la base de datos [2]. La idea que subyace al método *PI* es que cada objeto de la base de datos se representa por la permutación de un conjunto de referencias ordenadas por distancia al objeto. La distancia entre objetos es sugerido por la distancia entre sus respectivas permutaciones. Más formalmente, sea $P = \{p_1, p_2, \dots, p_k\}$ y $u \in U$, se define Π_u como una permutación de $(1 \dots k)$ tal que, para todo $1 \leq i < k$ se cumple que $d(p_{\Pi_u(i)}, u) < d(p_{\Pi_u(i+1)}, u)$ o $d(p_{\Pi_u(i)}, u) = d(p_{\Pi_u(i+1)}, u)$ y $\Pi_u(i) < \Pi_u(i+1)$.

En [13] se presenta *The Prefix Permutation Index (PP-Index)*, que representa cada objeto indexado como un prefijo corto de permutación. A diferencia de los demás índices basados en permutación que utilizan permutaciones para estimar el orden de la distancia real de los objetos indexados con respecto a una consulta, el *PP-Index* utiliza los prefijos de permutación para recuperar rápidamente un conjunto de tamaño razonable de objetos candidatos que probablemente están a corta distancia del objeto de consulta para luego dejar que la función de distancia seleccione los mejores elementos de entre los candidatos.

En [14] se presenta *Metric Inverted Index (M-Index)* que simplifica el cálculo de distancia entre permutaciones. Esta

técnica considera que cuando dos objetos están muy cerca uno del otro *perciben el mundo alrededor de ellos* de la misma manera, por tanto puede utilizarse una medida de disimilitud de dicha percepción del entorno de dos objetos. En [19] se presenta un estudio sobre las propiedades del espacio particionado de los índices basados en permutación.

III. SOLUCIÓN PROPUESTA

En este trabajo se presenta una técnica de indexación basada en permutaciones de los índices del vector de intensidad de píxeles que representa una imagen. El método propuesto, denominado **Emiap**, es una alternativa para el proceso de búsqueda en bases de datos de miles de imágenes y se implementa en combinación con el método PCA, que reduce la dimensionalidad de los datos.

Se plantean dos implementaciones de Emiap: *Emiap simple (emiap_s)* y *Emiap con combinación convexa (emiap_c)*.

Tanto la implementación de **emiap_s** como la de **emiap_c**, están divididas en dos fases: entrenamiento y reconocimiento. En la fase de entrenamiento se encuentran los *eigenectores* de las imágenes de la base de datos para luego indexarlas. En la fase de reconocimiento, la imagen de consulta se proyecta al subespacio PCA, se determina su índice y se procede a buscar en la base de datos la imagen más similar a la imagen de consulta. Durante la búsqueda se comparan los índices para determinar los candidatos a comparar directamente con la consulta.

En la Figura 1 se observa la implementación del método PCA para 8 imágenes faciales extraídas de la base de datos FERET. Primeramente se leen las imágenes faciales para representarlas en el espacio vectorial y se pre-procesan. En el pre-procesamiento, las imágenes se escalan a un tamaño determinado, se convierten a escala de grises, se ecualiza el histograma. Luego se realiza el cálculo de PCA sobre las imágenes pre-procesadas. En este trabajo, el método PCA se implementa mediante la librería OpenCV [6].

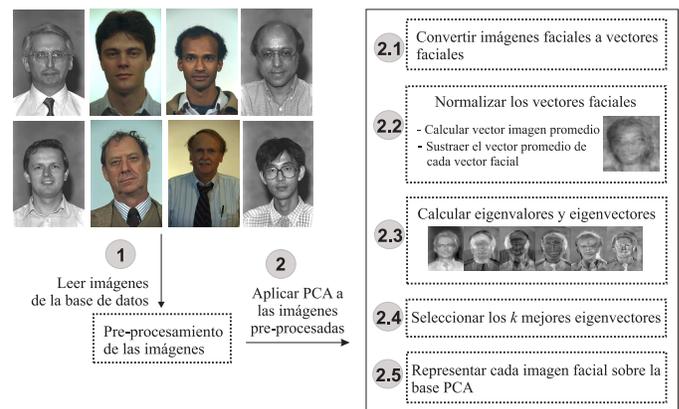


Figura 1: Esquema de la implementación del método PCA

III-A. Medidas de similitud aplicadas a imágenes faciales

Las funciones de distancia representan la manera de cuantificar la proximidad de los objetos en un dominio determinado.

Entre las métricas de similitud aplicadas a las imágenes en el presente trabajo se tienen las siguientes:

III-A1. Distancia de Minkowski: La distancia de Minkowski está conformada por toda la familia de funciones métricas denominadas métricas L_p . Estas distancias se definen sobre vectores de números reales x, y : $L_p[(x_1, \dots, x_n), (y_1, \dots, y_n)] = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$ [10]. En este trabajo se utilizan las distancias L_2 y L_∞ :

- Euclidiana

$$L_2 = d_2(x, y) = \sqrt{\sum_{i=1}^m |x_i - y_i|^2}, \quad (1)$$

- Máxima

$$L_\infty = d_\infty(x, y) = \max_{(i=1 \dots m)} |x_i - y_i| \quad (2)$$

III-A2. Distancia Mahalanobis:

$$d(x, y) = \sqrt{(x - y)^T C^{-1} (x - y)}, \quad (3)$$

donde x, y , son las variables aleatorias con la misma distribución de probabilidad y C es la matriz de covarianza. De este modo, las variables se ponderan según el grado de relación que exista entre ellas. Si la matriz de covarianza es la matriz identidad, la distancia Mahalanobis sería igual a la distancia euclidiana.

III-B. Medidas de similaridad entre permutaciones

Sea $\Pi_y = \{p_1, p_2, p_3, p_4, p_5, p_6\}$ la permutación de la consulta, y sea $\Pi_x = \{p_3, p_6, p_2, p_1, p_5, p_4\}$ la permutación de un elemento x de la base de datos, donde p_i forma parte del conjunto de ordenadores P . A continuación se definen las medidas de similaridad entre permutaciones utilizadas en este trabajo [9], [2]:

- Spearman's Rho

$$S_\rho(\Pi_y, \Pi_x) = \sqrt{\sum_{(i=1)}^k |\Pi_x^{-1}(p_i) - \Pi_y^{-1}(p_i)|^2} \quad (4)$$

El resultado final para el ejemplo es $S_\rho(\Pi_y, \Pi_x) = \sqrt{34} = 5,83$

- Spearman's Footrule

$$F(\Pi_x, \Pi_y) = \sum_{(i=1)}^k |\Pi_x^{(-1)}(p_i) - \Pi_y^{(-1)}(p_i)| \quad (5)$$

El resultado final para el ejemplo es $F(\Pi_x, \Pi_y) = 12$

- Kendall Tau

$$K(\Pi_x, \Pi_y) = \sum_{\{p_i, p_j\} \in P} K_{p_i, p_j}(\Pi_x, \Pi_y) \quad (6)$$

Se define por la Ecuación 6 [11], donde para cada par $\{p_i, p_j\} \in P$, si p_i y p_j están en el mismo orden en Π_y y Π_x , (esto es $\Pi_x^{(-1)}(p_i) < \Pi_x^{(-1)}(p_j) \iff \Pi_y^{(-1)}(p_i) < \Pi_y^{(-1)}(p_j)$) entonces $K_{p_i, p_j}(\Pi_x, \Pi_y) = 0$; si no $K_{p_i, p_j}(\Pi_y, \Pi_x) = 1$. El resultado final para el ejemplo es $K(\Pi_x, \Pi_y) = 15$.

III-C. Emiap simple

Emiap_s se basa en la propuesta de *Permutation Index (PI)*, en la cual los índices están dados por permutaciones. Las permutaciones representan el orden en el que los elementos de un conjunto de imágenes P son percibidos por cada objeto de la base de datos que será indexada. En el presente trabajo se evalúa el método *PI* con la selección aleatoria de los elementos del conjunto P [1] y se propone la implementación de heurística en la selección del conjunto P . También se explora el comportamiento del método aplicando otras métricas de similitud, además de la euclidiana utilizada en la propuesta de [8], para calcular las distancias de los elementos de P a cada objeto de la base de datos. La similaridad entre permutaciones se determina a través de las métricas definidas en la Subsección III-B. La imagen buscada y los candidatos resultantes se comparan directamente a través de las distancias definidas en la Subsección III-A.

III-C1. Selección del conjunto ordenador: El conjunto ordenador se compone de c imágenes $P = \{p_1, \dots, p_c\}$, donde el valor de c se determina empíricamente.

En el presente trabajo se evalúa la selección aleatoria y la selección mediante heurística.

Se incluye la selección aleatoria para establecer un marco comparativo del presente trabajo con la propuesta de [8]. En este método se seleccionan aleatoriamente c vectores que representan imágenes proyectadas al subespacio PCA y que conformarán el conjunto P . La heurística propuesta consiste en seleccionar elementos del conjunto P minimizando el valor de S_ρ respecto a los elementos que se van seleccionando. Con la minimización de S_ρ se busca obtener elementos que estén muy cerca unos de otros, con lo cual las permutaciones serán más sensibles a pequeños cambios de posición en el espacio [9].

La heurística se implementa mediante **algoritmo genético simple**. Los individuos son vectores que representan imágenes proyectadas en el subespacio PCA. La población inicial se genera aleatoriamente, y el tamaño de la misma se determina arbitrariamente. Cada individuo es evaluado mediante la función que se desea minimizar $f(x) = \sqrt{\sum_{(i=1)}^c |x_{1i} - x_{2i}|^2}$, donde x_1 y x_2 inicialmente son aleatorios y c es la longitud de los vectores. Luego x_2 es reemplazado por el individuo seleccionado en la iteración anterior. El proceso se repite hasta tener todos los elementos del conjunto P .

En la función de selección del algoritmo genético simple se implementa la selección proporcional estándar para problemas de minimización, incorporando modelo elitista con el fin de asegurar que el mejor individuo sobreviva. Se realiza una mutación aleatoria uniforme en el cual una variable seleccionada se reemplaza por un valor aleatorio. Luego se computa la función de evaluación para obtener el fitness de cada individuo y se guarda el mejor individuo de la generación anterior.

III-C2. Proceso de indexación: El proceso de indexación es la ordenación relativa de las imágenes de la base de datos. Las imágenes proyectadas al subespacio PCA conforman el universo de datos U . Entonces, dada una base de datos de

imágenes X , $X \subseteq U$, y un conjunto ordenador P , $P \subseteq U$, tal que $\forall x_i \in X$, con $i = \{1, \dots, |X|\}$ y $|X| = n$, se calcula su orden respecto al conjunto ordenador P seleccionado en un proceso anterior. Esto es $\forall p_j \in P$, con $j = \{1, \dots, |P|\}$ y $|P| = c$, se halla la distancia $x'_i = \{d(x_i, p_j)\}$ mediante métricas definidas en la Subsección III-A. El vector de las posiciones de los elementos de P resultantes en la ordenación ascendente de x'_i , representa la permutación del objeto x_i de la base de datos y es el que finalmente será utilizado como índice Π_{x_i} de x_i en las consultas a la base de datos. El esquema de indexación se ilustra en la Figura 2.

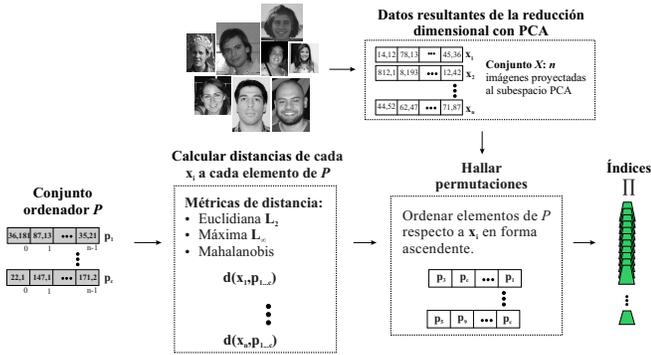


Figura 2: Proceso de indexación de la base de datos mediante emiap_s

La ordenación de x'_i se lleva a cabo mediante el algoritmo *Quicksort* que en [8] ha permitido obtener mejores resultados. En el presente trabajo se explora el comportamiento de una versión paralelizada de *Quicksort* mediante la librería OpenMP.

III-C3. Proceso de búsqueda: Como primer paso la imagen buscada, y , se proyecta al subespacio PCA obteniéndose el vector v . Luego se calcula la distancia euclidiana entre v y cada objeto del conjunto P , es decir, $\forall p_j \in P$ con $j = \{1, \dots, |P|\}$ y $|P| = c$, se halla $v' = \{d(v, p_j)\}$. Las posiciones de los elementos de P en la ordenación ascendente de v' es la permutación de v que representa el índice Π_v de v . Seguidamente se calculan las distancias entre el índice de la imagen buscada y los índices de cada elemento de la base de datos que han sido construidos durante el proceso de indexación. Dicho cálculo se realiza mediante las métricas presentadas en la Subsección III-B. Luego se ordena el vector de distancias entre permutaciones para identificar a los candidatos que posteriormente serán comparados con la imagen buscada aplicando las métricas definidas en la Subsección III-A. En la Figura 3 se esquetiza el proceso de búsqueda.

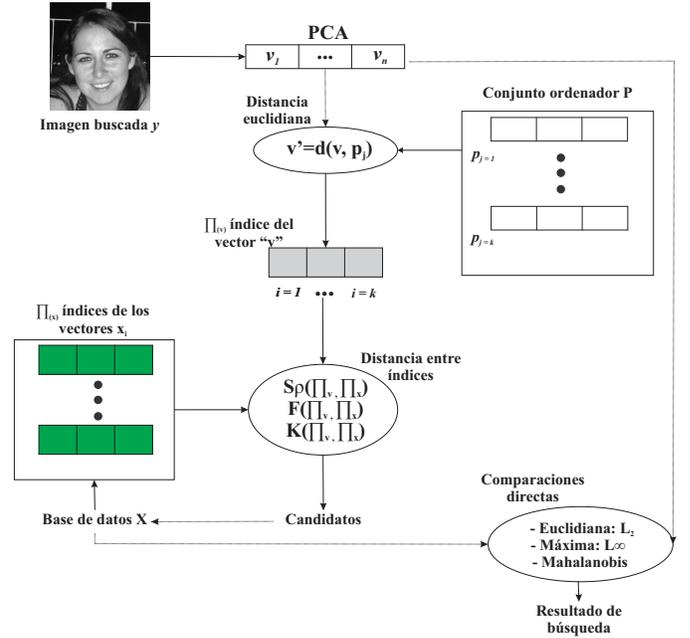


Figura 3: Esquema de búsqueda con indexación mediante emiap_s

III-D. Emiap con combinación convexa

Emiap_c consiste en la implementación de algoritmos de aproximación paramétrica [3] en el proceso de indexación que luego permitirán establecer la similitud entre vectores de características de las imágenes faciales.

El método de indexación presentado en esta sección se basa en la propuesta de [12], en la cual un vector $x = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^n$ se representa como una combinación convexa del par $(s(x), \sigma(x))$, donde $s(x)$ es la versión ordenada de x y $\sigma(x)$ es la permutación de los índices $\{1, 2, \dots, n\}$ que consigue el ordenamiento de x . Un ejemplo de esto se observa en la Figura 4 donde tras la ejecución de un proceso de ordenación, las coordenadas del vector x quedan ordenadas en forma creciente, al construir $\sigma(x)$, en caso de empates se tiene en cuenta el orden de aparición, obsérvese que $x_2 = x_4 = 11,3$.



Figura 4: Descomposición del vector x en el par $(s(x), \sigma(x))$

Un subconjunto $X \subset \mathbb{R}^n$ es convexo si para cualesquiera puntos $x, y \in X$ y cualquier $\lambda \in [0, 1]$ se tiene que $\lambda x + (1 - \lambda)y \in X$ [21].

La distancia entre dos vectores se aproxima por una combinación convexa de la medida de la forma de los vectores y de las magnitudes de sus coordenadas. Siendo $x, y \in \mathbb{R}^n$ vectores reales n -dimensionales, una combinación convexa se define por la Ecuación 7, donde $d(x, y)$ es la distancia euclidiana entre

x y y , denominada como la parte de la magnitud (o parte simétrica) y $\Pi_{(x,y)}$ es la medida entre las permutaciones $\sigma(x)$ y $\sigma(y)$, denominada como la parte de la forma [3].

$$\lambda d_{(x,y)} + (1 - \lambda)\Pi_{(x,y)} \quad (7)$$

III-D1. Aproximación de la parte de la magnitud: El cálculo de la distancia $d_{(x,y)}$, que corresponde a la parte de la magnitud en la Ecuación 8, se realiza mediante una aproximación de la distancia euclídea definida por la Ecuación 7, donde $\varphi_p(x)$, $\varphi_p(y)$ son potencias simétricas de x e y respectivamente y $\langle x, y \rangle = x_1y_1 + x_2y_2 + \dots + x_ny_n$ es el producto interno estándar [3].

$$\|x - y\| = \sqrt{\varphi_2(x) + \varphi_2(y) - 2\langle x, y \rangle} \quad (8)$$

Primeramente se calculan las potencias simétricas mediante la Ecuación 9, el valor de p se define por m , es decir, si $m = 2$, entonces $p = 2$ y se calculan las potencias simétricas $\varphi_1(x)$ y $\varphi_2(x)$ para cada vector.

$$\varphi_p(x) = x_1^p + x_2^p + \dots + x_n^p \quad (9)$$

Luego, usando los valores reales $\varphi_p(x)$ y $\varphi_p(y)$ precalculados para cada vector $x = (x_1, x_2, \dots, x_n)$ e $y = (y_1, y_2, \dots, y_n)$, se busca la estimación del producto interno $\langle x, y \rangle$ mediante la aproximación de la m -ésima potencia de la forma [3]:

$$\langle x, y \rangle^m \approx b_1\varphi_1(x)\varphi_1(y) + \dots + b_m\varphi_m(x)\varphi_m(y) \quad (10)$$

para una n muy grande, donde $|x| = |y| = n$ y b_i son constantes universales elegidos independientemente de x e y . En el presente trabajo los valores para las constantes b_i han sido tomados de [12].

III-D2. Aproximación de la parte de la forma: El cálculo de distancia entre permutaciones, $\Pi_{(x,y)}$, es costoso considerando la dimensión real n de cada vector, por ello, en [3] se consideran las permutaciones *bin-score*, que permite encontrar una aproximación de la permutación real, con dimensiones inferiores, y que luego será usada para encontrar las medidas de distancias sin mucha desviación del valor real.

El i -ésimo *bin* se describe por d índices consecutivos: $i_1 = (i-1)d+1$, $i_2 = (i-1)d+2 \dots i_d = (i-1)d+d$. El score de este *bin* es la suma de $\rho_{i_1}, \rho_{i_2}, \dots, \rho_{i_d}$. Y así se construye otra versión ρ sobre $\{1, 2 \dots d\}$ reemplazando 1 por el elemento más pequeño en el score calculado, 2 por el siguiente más pequeño y así siguiendo. En caso de empate, se incrementan los índices de izquierda a derecha.

Por ejemplo, para $n = 8$, la permutación *bin-score* de $\rho = 5, 8, 2, 6, 4, 3, 1, 7$ para $d = 4$ y $d = 2$ se obtienen de los scores 13, 8, 7, 8 y 21, 15 como las permutaciones 4, 2, 1, 3 y 2, 1 respectivamente. De esta manera una permutación de n -dimensional se aproxima por una permutación d -dimensional.

III-D3. Proceso de indexación: En **emiap_c** una parte del índice es un vector de permutación, utilizado en el cálculo de la aproximación de la forma, y la otra parte son números reales, utilizados en el cálculo de la aproximación de la magnitud.

Cada imagen de la base de datos representada en el subespacio PCA se ordena en sus coordenadas ascendentes. Con el ordenamiento se obtienen las componentes $s(x_i)$, utilizadas en el cálculo de la parte de la magnitud, y $\sigma(x_i)$, utilizadas en el cálculo correspondiente a la parte de la forma.

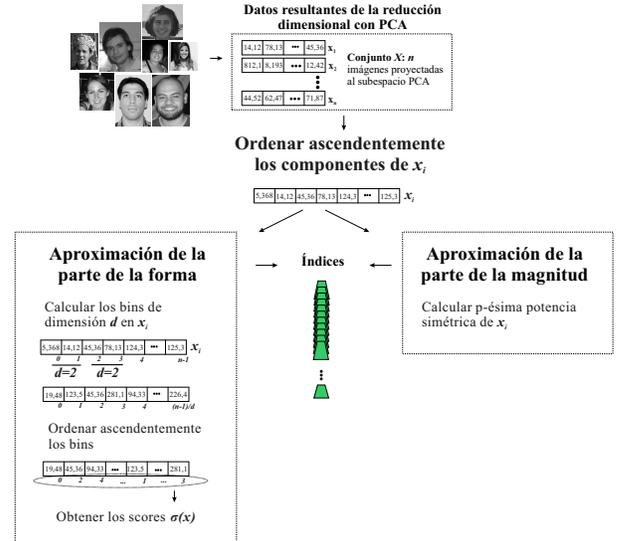


Figura 5: Proceso de indexación de la base de datos mediante **emiap_c**

Para el cálculo de la *aproximación de la parte de la magnitud* entre los vectores del conjunto X y el vector de consulta se requieren los valores de la p -ésima potencia simétrica $\varphi_p(x_i)$, por tanto estos valores reales se almacena como parte del índice del vector x_i . La cantidad de potencias simétricas halladas está determinada por la definición del valor de m .

Para la *aproximación de la parte de la forma* del vector x_i se aplica el método *bin-score* [3] a las componentes del mismo obteniéndose así los *scores* $\sigma(x_i)$, tras calcular los *bins*. Los *scores* $\sigma(x_i)$ son las nuevas permutaciones y también forman parte del índice del vector x_i . El esquema de indexación se ilustra en la Figura 5.

III-D4. Proceso de búsqueda: En el presente trabajo se evalúa el comportamiento del esquema de combinación convexa definiendo arbitrariamente los valores de m , definida en la Ecuación 9, para el cálculo de la m -ésima potencia que aproxima el producto interno entre los vectores de la base de datos x_i y el vector de consulta v .

En primer lugar el vector imagen transformado en el subespacio PCA es ordenado en sus coordenadas ascendente. Del proceso de ordenación se tienen las componentes $s(v)$ y $\sigma(v)$. Se aplica el método *bin-score* al vector de consulta para obtener la nueva permutación $\sigma(v)$ y se calcula la p -ésima

potencia simétrica a las componentes del vector de consulta. El esquema de búsqueda se ilustra en la Figura 6.

Para calcular la distancia aproximada entre los vectores de la base de datos y el vector de consulta se aplica la Ecuación 7. Los valores de λ son evaluados en el intervalo $0,0 \leq \lambda \leq 1$. La distancia entre las aproximaciones de la parte de la forma, $\Pi(x, v)$, se calcula mediante las métricas definidas en III-B. La distancia entre las aproximaciones de la parte de la magnitud, $d(x, v)$, se aplica la Ecuación 8, para lo cual se utilizan las potencias simétricas pre-calculadas y que forman parte de la estructura del índice.

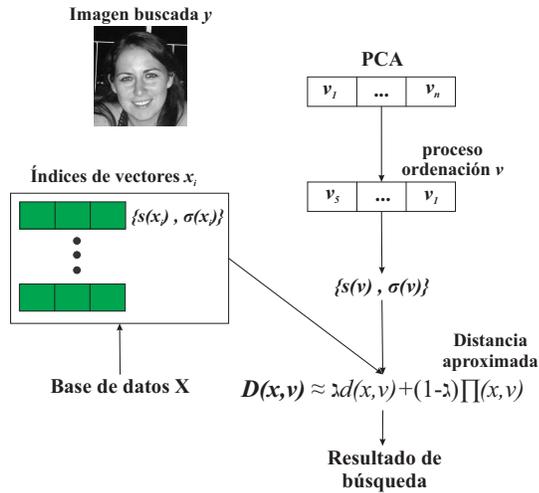


Figura 6: Esquema de búsqueda con indexación mediante emiap_c

Las distancias entre índices, resultantes de la Ecuación 7, son ordenados ascendentemente para identificar a los candidatos que luego serán comparados directamente con la imagen buscada mediante las métricas definidas en III-A.

IV. RESULTADOS EXPERIMENTALES

Los experimentos fueron realizados sobre la base de datos FERET. Tanto en la fase de entrenamiento como de reconocimiento se manejan imágenes tomadas desde distintos ángulos. Se evalúan los tiempos de construcción de índices y los porcentajes de aciertos. Al igual que en [8], en el presente trabajo la tasa de aciertos en una consulta se mide por:

$$\%Aciertos = \frac{|R_{alg} \cap R_{cor}|}{|R_{cor}|} \times 100,$$

donde:

R_{alg} es el conjunto de respuesta entregada por el algoritmo.

R_{cor} es el conjunto de respuesta correcta.

$|R_{cor}| = K$ la cantidad de elementos evaluados.

IV-A. Evaluación durante la construcción de índices

Los siguientes experimentos muestran que, al igual que los métodos de indexación presentados en [8], [13], las implementaciones de **Emiap** tienen una proporción casi lineal del coste de tiempo de indexación con respecto al tamaño de la base de datos y al tamaño de los índices. Las Figuras 7a y

7b muestran que el tiempo de indexación es mejorado con la implementación del método de ordenación quicksort en cuatro hilos de ejecución mediante OpenMP.

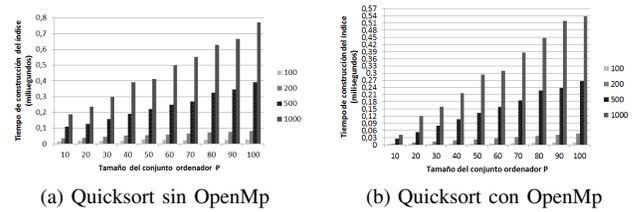


Figura 7: Construcción de índice emiap_s

En la Tabla I se presentan los tiempos de indexación y la cantidad de espacio en disco que ocupan los índices tras ser construídos con las tres medidas de similitud aplicadas en este trabajo para la comparación de cada objeto de la base de datos y cada objeto del conjunto ordenador P .

TABLA I: Tiempo de indexación de emiap_s

Algoritmo	Disco (KB)	Tiempo de indexación (ms)
Euclidiana L_2	81	0,463
Máxima L_∞	80	0,388
Mahalanobis	79	0,581

En la Figura 8 se muestra el tiempo de construcción de índices mediante **emiap_c**. Se evalúa la eficiencia en la indexación de la base de datos con 100, 200, 500 y 1000 elementos utilizando distintos tamaños de *bins* (donde d es el tamaño de bin).

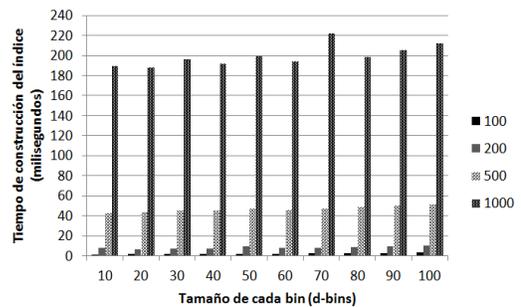


Figura 8: Construcción de índice emiap_c

En la Tabla II se presentan los tiempos de indexación y la cantidad de espacio en disco que ocupan los índices construídos con el algoritmo de combinación convexa. Se observa que cuanto mayor sea la dimensión d de cada bin se tiene índices que ocupan menos espacio en disco pero con mayor tiempo de indexación.

TABLA II: Tiempo de indexación de emiap_c

Dimensión d -bins	Disco (KB)	Tiempo de indexación (ms)
5	55	250
10	45	320
15	40	387
20	35	420

IV-B. Evaluación durante las búsquedas

En la Figura 9 se muestra el tiempo de respuesta en una búsqueda múltiple aplicando la indexación mediante **emiap_c**. Para este experimento la cantidad de elementos de la base de datos que se compara directamente se mantiene fija en 30. Se exploran distintos tamaños de *bins* y las permutaciones (*scores*) resultantes se comparan mediante F .

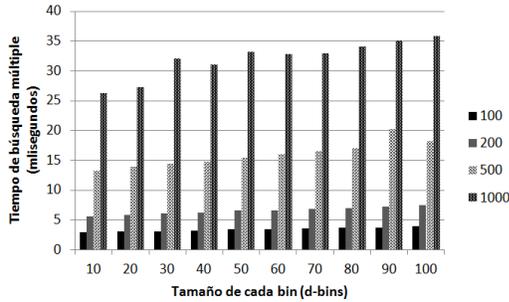


Figura 9: Búsqueda múltiple usando emiap_c

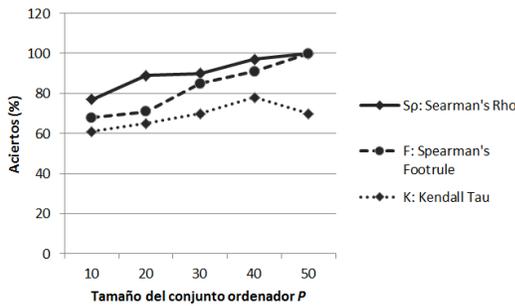


Figura 10: Porcentaje de aciertos en emiap_s con $S\rho$

En la Figura 10 se compara el impacto de las medidas de similaridad entre permutaciones utilizadas en la implementación de **emiap_s** en la tasa de aciertos. Este experimento se realizó sobre una base de datos de tamaño $N = 1500$, la selección de los objetos del conjunto ordenador P es mediante la heurística y la cantidad de candidatos comparados directamente mediante la distancia *Mahalanobis* se determina en un 10% del tamaño de la base de datos. Las medidas de similaridad entre permutaciones¹ también han sido evaluadas en [8], trabajo en el cual se ha concluido que $S\rho$ y K tienen el mismo desempeño, pero el primero es menos costoso de calcular. Por otro lado, F tiene una evaluación menos costosa que los otros dos pero su desempeño en cuanto a proximidad es más pobre. Sin embargo, en el presente trabajo, se muestra que $S\rho$ y F arrojan mejores resultados que K , por ejemplo para $|P| = 50$ tanto en $S\rho$ como F se obtiene un 100% de aciertos.

Obsérvese en la Figura 11 los porcentajes de aciertos obtenidos mediante la indexación por **emiap_c** usando distintas

¹Spearman's Rho ($S\rho$), Spearman's Footrule (F) y Kendall Tau (K).

configuraciones. La similitud entre el índice² del elemento buscado y de los existentes en la base de datos se mide mediante la métrica F . En este experimento se mantiene constante la cantidad de elementos de la base de datos que se compara directamente en 30. Las pruebas se realizan sobre base de datos con 100, 200, 500 y 1000 elementos utilizando distintos tamaños de *bins* (donde d es el tamaño de bin).

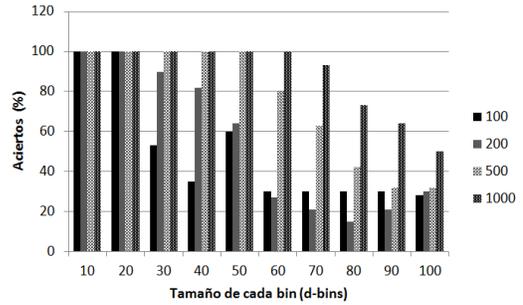


Figura 11: Porcentaje de aciertos aplicando emiap_c

En la Tabla III se muestra el porcentaje de acierto de **emiap_c** utilizando como medida de similitud entre las permutaciones (*scores*): $S\rho$, F y K utilizadas en la parte de la forma de la Ecuación 7. Este experimento se realiza con $N = 1500$ imágenes, $b = 10$ y se comparan directamente los primeros 100 elementos de la base de datos cuyos índices distan menos del índice del elemento buscado según las diferentes medidas de similitud aplicadas.

TABLA III: Aciertos en emiap_c con distintas medidas de similitud entre permutaciones

Medida de similitud aplicadas en la parte de la forma	Aciertos (%)
Spearman's Rho	96
Spearman's Footrule	97
Kendall Tau	75

En la Tabla IV se muestra el impacto del valor de λ de la Ecuación 3.10 en la tasa de aciertos, donde λ es el peso que se le da a cada parte de la combinación convexa: a la parte de la magnitud y a la parte de la forma. Este experimento se realiza sobre $N = 1500$ imágenes pertenecientes a 187 personas, la dimensión de cada bins es $d = 8$, con $m = 2$ y como medida de similitud entre las permutaciones (*scores*) se utiliza F .

²Las permutaciones o *score* resultantes del cálculo de *bins*.

TABLA IV: Aciertos en emiap_c para distintos valores de λ de la Ecuación 3.10

λ	Spearman's Rho	Spearman's Footrule	Kendall Tau
0.0	45	58	18
0.1	49	62	19
0.2	51	73	19
0.3	69	89	46
0.4	94	96	74
0.5	88	90	70
0.6	76	82	65
0.7	64	71	56
0.8	50	62	48
0.9	44	54	44
1.0	36	47	30

En la Tabla V se muestra el impacto del valor de m en la tasa de aciertos. Este experimento se realiza con $N = 1500$ imágenes pertenecientes a 187 personas, $d = 10$, el valor de λ de la Ecuación 7 utilizado es 0,4 y como medida de similitud entre las permutaciones (*scores*) se utiliza F . Para la mayoría de las aplicaciones basadas en búsquedas por proximidad, recuperar el 90 % de las respuestas es un resultado aceptable [13]. En el caso del algoritmo con combinación convexa es necesario revisar un 60 % de la base de datos para obtener ese porcentaje. Se observa que a medida que la dimensión crece es necesario revisar una fracción más grande de la base de datos para obtener la misma precisión. En ambos algoritmos ocurre lo mismo, aunque no en la misma proporción. Por otro lado, es posible revisar un porcentaje menor de la base de datos si se incrementa el número de elementos en el conjunto ordenador $|P|$.

TABLA V: Aciertos en emiap_c para distintos valores de m

m	Aciertos (%)
2	90
3	93
4	98

En la Figura 12 se comparan las implementaciones presentadas en este trabajo con otros métodos del estado de arte. Se observan los porcentajes de comparaciones directas entre la consulta y los objetos de la base de datos utilizando PI, PP-Index, **emiap_c** y **emiap_s** para obtener un porcentaje de acierto por encima del 90 %. Este experimento se realiza con $N = 1500$ imágenes pertenecientes a 187 personas y la métrica utilizada para hallar las distancias entre permutaciones es el $S\rho$. En el caso de PI, la selección de los objetos del conjunto P se realiza en forma aleatoria, el tamaño del conjunto P es $|P| = 100$, las distancias entre los objetos de la base de datos y los objetos de P, al igual que las distancias entre la imagen buscada y los candidatos encontrados, se calculan mediante la distancia euclidiana. En el caso de PP-Index, también el tamaño del conjunto P es $|P| = 100$, las distancias entre los objetos de la base de datos y los objetos de P, al igual que las distancias entre la imagen buscada y los candidatos encontrados, se calculan mediante la distancia euclidiana. En el caso de **emiap_s** la selección de los objetos del conjunto P, $|P| = 100$, se realiza mediante la heurística

presentada en este trabajo y las distancias entre los objetos de la base de datos y los objetos de P, las distancias entre la imagen buscada y los candidatos encontrados, se calculan mediante la distancia Mahalanobis. En el caso de **emiap_c** se utilizan los siguientes valores $\lambda = 0,3$, $d = 20$ y $m = 2$. **Emiap_c** es el que realiza más comparaciones directas entre la imagen buscada y las almacenadas en la base de datos.

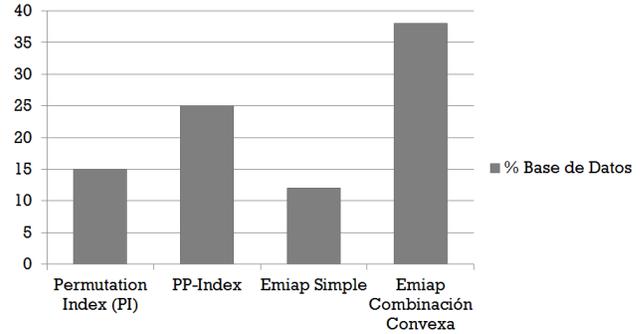


Figura 12: Porcentaje de base de datos comparada directamente

En la Figura 13, se seleccionan 128 elementos componentes del conjunto ordenador, en forma aleatoria como en [8] y mediante la heurística presentada en este trabajo. Examinando el 10 % de una base de datos con 1000 imágenes se recupera el 99 % de los resultados correctos con el algoritmo de **emiap_s** en el que se utiliza la heurística para seleccionar los elementos del conjunto ordenador P. Para la solución de combinación convexa el tamaño de bins se determina en 20 y en la Tabla VI, sin embargo, se observa que el algoritmo **emiap_c** requiere comparar el 25 % de la base de datos para recuperar el mismo porcentaje.

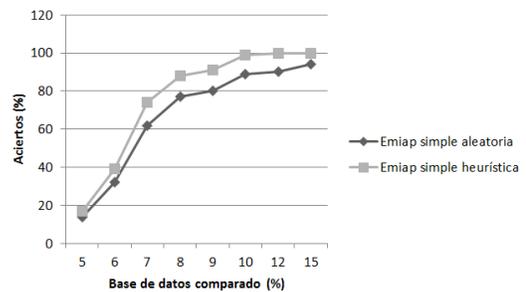


Figura 13: Porcentaje de la base de datos comparada directamente con emiap_s

TABLA VI: Porcentaje de la base de datos comparada directamente mediante emiap_c

Base de datos comparada (%)	Aciertos (%)
26	65
28	74
30	83
32	85
35	99
38	100

Según los experimentos realizados, en *emiap_s*, el porcentaje de comparaciones directas entre la consulta y los candidatos hallados en la base de datos es similar empleando cualquiera de los métodos de selección de los elementos del conjunto ordenador P (aleatoria o heurística). Sin embargo, el porcentaje de aciertos es mayor si los elementos de P se eligen mediante la heurística planteada en este trabajo.

V. CONCLUSIONES

En el presente trabajo se han desarrollado técnicas complementarias para la fase de clasificación de un proceso de reconocimiento facial, que maneja un espacio de búsqueda compuesto por miles de imágenes. Se utilizan diferentes métodos para cada fase del reconocimiento. Durante el pre-procesamiento de las imágenes se implementan métodos de escalado y ecualización de histograma. En la extracción de características se aplica el método PCA (*Principal Analysis Component*) que permite contar con vectores de características de baja dimensionalidad capaces de dar buenos resultados en reconocimiento. Como clasificador se utiliza la búsqueda por similaridad mediante el k -NN (*k-Nearest Neighbours*).

El objetivo de este trabajo es recorrer solo una pequeña proporción de la base de datos sin comprometer la alta tasa de aciertos, para lo cual se propone una nueva técnica para la búsqueda por similaridad en espacios métricos, denominada Emiap. Se introducen dos implementaciones de Emiap: Emiap simple (*emiap_s*) y Emiap con combinación convexa (*emiap_c*).

Los experimentos realizados evalúan las etapas más relevantes del funcionamiento global del sistema y permiten medir la eficiencia y eficacia de los métodos de indexación propuestos en este trabajo. Las pruebas han sido realizadas sobre la base de datos FERET.

Las medidas de similitud entre las imágenes de consulta y los candidatos encontrados tienen una fuerte incidencia en el rendimiento en términos de tasa de reconocimiento como de velocidad de procesamiento. Los experimentos muestran que la distancia de *Mahalanobis* ofrece una mejor tasa de acierto, sin embargo, en términos de eficiencia la distancia *máxima* presenta mejor desempeño.

Las pruebas realizadas en el marco de la medición de la eficacia revelan que las técnicas basadas en permutaciones son muy buenas para predecir la relevancia de un objeto en una consulta. En la implementación de *emiap_s*, el aporte de este trabajo es la selección de elementos del conjunto P mediante heurística, pues se demuestra que de este modo aumenta el nivel de tasa de aciertos respecto a la selección de P realizada en [8]. También se demuestra que *emiap_c* es una alternativa competitiva a otras técnicas existentes en el estado de arte en relación a la tasa de aciertos.

En cuanto al tiempo de construcción de los índices, se encuentra que ambas implementaciones del presente trabajo tienen un coste lineal respecto a la cantidad de elementos existentes en la base de datos, al igual que otras técnicas presentadas en [8], [13]. En *emiap_c* se realizan más cálculos algebraicos que en *emiap_s*, pero en este último se realizan

ordenaciones, mediante *Quicksort*, de los elementos de P por cada elemento de la base de datos.

El espacio en disco que ocupan los índices generados mediante *emiap_c* puede alcanzar un tamaño 50% menor al espacio ocupado por los índices generados mediante *emiap_s*, manteniendo una tasa de aciertos por encima del 90%.

El tiempo de respuesta en *emiap_s* es lineal respecto a la cantidad de elementos que componen el conjunto P y la cantidad de elementos existentes en la base de datos, en *emiap_c*, sin embargo, es lineal respecto la dimensión de cada *bins* y a la cantidad de elementos existentes en la base de datos.

En definitiva, la *emiap_c*, ocupa menos espacio en disco y, por tanto, al momento de la búsqueda ocupa menos espacio en memoria. Además presenta una alta tasa de aciertos, con lo cual se convierte en una alternativa competitiva en bases de datos de millones de elementos. *Emiap_s* es de implementación más sencilla y se ha demostrado en este trabajo que con selección mediante heurística de elementos de P , presenta tasas de aciertos más elevadas que con la selección aleatoria, lo que también la convierte en una alternativa de indexación competitiva respecto a otros métodos del estado de arte.

V-A. Trabajos futuros

Con el fin de continuar y mejorar lo estudiado en el presente trabajo, se proponen diferentes líneas de trabajo:

- Utilización del método de indexación Emiap en otros dominios de aplicación.
- Mejorar la definición de heurística para la selección de elementos del conjunto ordenador en *emiap_s*.
- Desarrollo y evaluación de nuevas técnicas que permitan el funcionamiento en tiempo real. Una de las variantes del algoritmo PCA es el *Linear Discriminant Analysis* (LDA) [22], [23], también conocida como *Fisherfaces*, que permite obtener mayor robustez frente a cambios de iluminación y tiene un coste computacional similar al de PCA.
- Implementación de paralelismo en el proceso de búsqueda de Emiap. las propiedades de procesamiento paralelo de Emiap hace posible distribuir el proceso de búsqueda con el fin de mejorar el tiempo de respuesta del sistema manteniendo su nivel de eficacia.
- Detección y reconocimiento de múltiples rostros. Una de las funciones que se beneficiarían directamente de la implementación de paralelismo en el sistema es la del funcionamiento para múltiples rostros en la misma imagen ya que utilizando un proceso para cada rostro, los tiempos no serían muy superiores al funcionamiento con un rostro.

REFERENCIAS

- [1] Edgar Chavez, Karina Figueroa, Gonzalo Navarro. Indexación Efectiva de Espacios Métricos usando Permutaciones, Tesis Doctor en Ciencias (Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, 2007).
- [2] Edgar Chavez, Karina Figueroa, Gonzalo Navarro, 2008. Effective Proximity Retrieval by Ordering Permutations, IEEE Trans. Pattern Anal. Mach. Intell., 30(9):1647-1658

- [3] Egecioglu, Ömer, Parametric Approximation Algorithms for High-Dimensional Euclidean Similarity, in Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (London, UK, UK: Springer-Verlag, 2001), pp. 79–90.
- [4] Lei Zhang, Yongdong Zhang, Jinhui Tang, Xiaoguang Gu, Jintao Li, and Qi Tian. 2013. Topology preserving hashing for similarity search. In Proceedings of the 21st ACM international conference on Multimedia (MM '13).
- [5] Phillips, P. Jonathon and Moon, Hyeonjoon and Rizvi, Syed A. and Rauss, Patrick J., The FERET Evaluation Methodology for Face-Recognition Algorithms, *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 10 (2000), pp. 1090–1104.
- [6] Kaehler, Adrian and Bradski, Gary, *Learning OpenCV. Computer Vision in C++ with the OpenCV Library* (O'Reilly Media, 2013).
- [7] Chávez, Edgar and Navarro, Gonzalo and Baeza-Yates, Ricardo and Marroquín, José Luis, Searching in Metric Spaces, *ACM Comput. Surv.* 33, 3 (2001), pp. 273–321.
- [8] Chavez Gonzalez, Edgar and Figueroa, Karina and Navarro, Gonzalo, Effective Proximity Retrieval by Ordering Permutations, *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 9 (2008), pp. 1647–1658.
- [9] Bustos, Benjamin and Navarro, Gonzalo, Probabilistic Proximity Searching Algorithms Based on Compact Partitions, *J. of Discrete Algorithms* 2, 1 (2004), pp. 115–134.
- [10] Vlastislav Dohnal, *Indexing Structures for Searching in Metric Spaces*. Ph.D. Thesis (Faculty of Informatics, Masaryk University, 2004).
- [11] Fagin, Ronald and Kumar, Ravi and Sivakumar, D., Comparing Top K Lists, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003), pp. 28–36.
- [12] Omer Egecioglu and Hakan Ferhatosmanoglu, Dimensionality Reduction and Similarity Computation by Inner Product Approximations, in *IEEE Transactions on Knowledge and Data Engineering (AAAI/MIT Press. He, 2000)*, pp. 714–726.
- [13] Esuli, Andrea, Use of Permutation Prefixes for Efficient and Scalable Approximate Similarity Search, *Inf. Process. Manage.* 48, 5 (2012), pp. 889–902.
- [14] Amato, Giuseppe and Savino, Pasquale, Approximate Similarity Search in Metric Spaces Using Inverted Files, in Proceedings of the 3rd International Conference on Scalable Information Systems (ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008), pp. 28:1–28:10.
- [15] Batur, A.U. and Flinchbaugh, B.E. and Hayes, M., III, A DSP-based approach for the implementation of face recognition algorithms, in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on vol. 2, (2003)*, pp. II-253-6 vol.2.
- [16] Shahrin Azuan Nazeer and Marzuki Khalid, PCA-ANN Face Recognition System based on Photometric Normalization Techniques, in *State of the Art in Face Recognition (Malaysia, 2009)*, pp. 71-86.
- [17] Eslava Ríos, Javier, *Reconocimiento Facial en Tiempo Real*. (Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, 2013).
- [18] Li, Stan Z. and Jain, Anil K., *Handbook of Face Recognition 2nd* (Springer Publishing Company, Incorporated, 2011).
- [19] Matthew Skala, Counting distance permutations, *Journal of Discrete Algorithms* 7, 1 (2009), pp. 49 - 61. Selected papers from the 1st International Workshop on Similarity Search and Applications (SISAP) 1st International Workshop.
- [20] Mohamed, Hisham, *Scalable approximate k-NN in multidimensional big data*. Tesis Doctor en Ciencias (Universidad de Geneva, 2014).
- [21] Cánovas, M.J.C. and Navarro, V.H. and Orts, M.S., *Optimización matemática aplicada. Enunciados, ejercicios y aplicaciones del mundo real con MATLAB: Enunciados, ejerci...* (Editorial Club Universitario, 2011).
- [22] Martínez, Aleix M., and Avinash C. Kak, PCA versus LDA. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.2 (2001): 228-233.
- [23] Carmen Virginia Gámez Jiménez. *Diseño y Desarrollo de un Sistema de Reconocimiento de Caras*, Tesis Ingeniería de Telecomunicación (Escuela Politécnica Superior, Universidad Carlos III de Madrid, 2009).