

6. Reconstrucción de imágenes ruidosas mediante técnicas basadas en la Factorización No Negativa de Matrices

6.1 Introducción

En este capítulo se presentan diferentes técnicas que van a permitir construir imágenes inmersas en un ruido aditivo gaussiano a partir de las técnicas *NMF* descritas anteriormente.

El desarrollo del estudio se puede dividir en dos partes bien diferenciadas: por un lado veremos las diferentes técnicas de reconstrucción de la imagen a partir de las bases obtenidas y por otra parte se estudiará la forma de eliminar el ruido que corrompe a las imágenes.

Por último, intentaremos comparar la reconstrucción basada en *NMF* con los resultados obtenidos al aplicar técnicas *ICA* y *PCA*.

6.2 Línea de acción

Seguidamente se procederá a detallar paso a paso el procedimiento que se ha seguido de cara a la realizar la reconstrucción de las imágenes ruidosas.

6.2.1 Descripción de datos

6.2.1.1 La base de datos

Para el desarrollo del algoritmo se cuenta con 400 fotografías de rostros de hombres y mujeres, procedentes de la base de datos *Cambridge ORL Face Database*. Dichas muestras se dividen en diez fotografías por cada una de las cuarenta personas que conforman la base.

A la hora de construir la base de datos se ha intentado en la medida de lo posible tener en cuenta variantes en los rostros que permitan enriquecer las bases que se obtendrán al aplicar el algoritmo *NMF*. Para ello se tomaron fotografías de las personas con diferentes gestos (sonriente, serio, triste, etc), portando objetos como gafas e incluso realizando pequeñas rotaciones de la cara que permiten observarla desde diferentes ángulos.

El tamaño de cada una de las cuatrocientas imágenes que conforman la base es de 112 píxeles de alto y 92 de ancho y están tomadas en escala de grises para así simplificar su procesamiento ya que tan sólo habrá que tener en cuenta un plano de color.

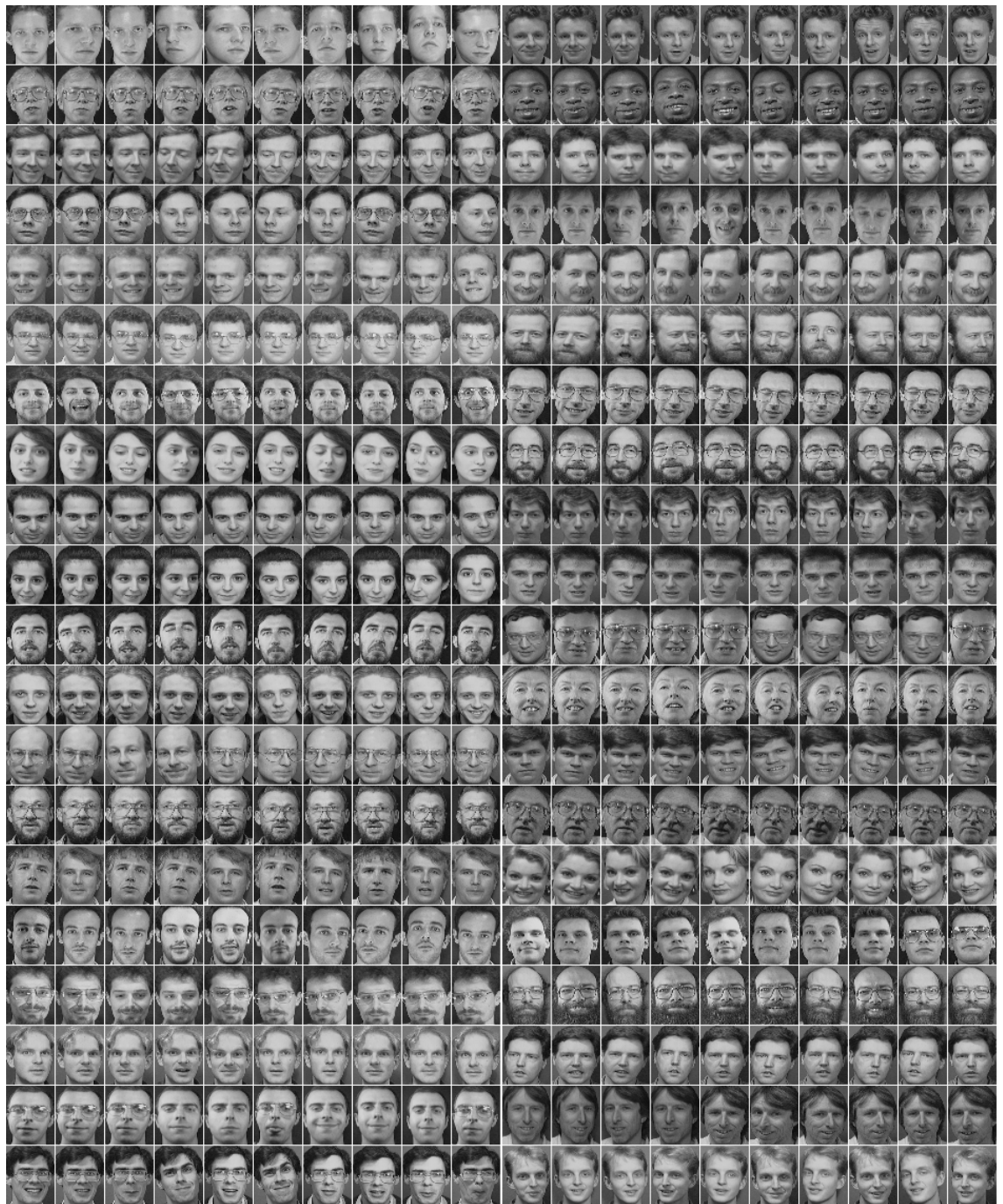


Figura 6.1 ‘Cambridge ORL Face Database’ empleada para la generación de bases mediante técnicas NMF. Esta base de datos consta de 400 imágenes representativas de rostros de cuarenta personas con rasgos bien diferenciados de cara a obtener la mayor variedad posible. Existen diez fotografías de cada una de las cuarenta personas, que han sido tomadas teniendo en cuenta posibles variantes en el rostro (variaciones en el gesto, uso de gafas, pequeñas rotaciones de la cara, etc). Cada imagen tiene un tamaño de 112 píxeles de alto y 92 de ancho.

6.2.1.2 La imagen a reconstruir

Como hemos comentado previamente, la imagen que se quiere reconstruir estará inmersa en ruido aditivo gaussiano. Dicha fotografía, al igual que las imágenes de la base de datos, tendrá un tamaño de 112 píxeles de alto por 92 de ancho y estará en escala de grises. Durante los diferentes experimentos que se vayan realizando alternaremos la reconstrucción de imágenes de la propia base (generalmente en una primera fase de prueba), con otros rostros que no guardan ninguna relación con la misma.



Figura 6.2 *Ejemplo de imagen a reconstruir perteneciente a la base. Como se puede observar en la figura, la imagen está inmersa en un ruido gaussiano que habrá que eliminar en el proceso de reconstrucción.*

6.2.2 Preprocesado de los datos

Dado que el tamaño de las imágenes con las que vamos a trabajar es considerablemente grande como para tratarlas directamente, se ha tomado la decisión de dividir las imágenes en regiones cuadradas de un tamaño mucho menor (contendrá ‘*tam*’ filas y ‘*tam*’ columnas), que denominaremos de aquí en adelante *patches*.

El preprocesado tanto de las imágenes que conforman la base de datos, como de la que se quiere construir es ligeramente diferente, como veremos en los siguientes apartados.

6.2.2.1 Preprocesado de la base de datos

Como paso previo a la aplicación del algoritmo *NMF* a la base de datos, hemos de construir una matriz que contendrá muestras representativas de las diferentes imágenes que conforman la base y que llamaremos $\mathbf{V}$.

Por otro lado, dado que el número de imágenes de la base es relativamente grande, si para conformar dicha matriz de entrenamiento $\mathbf{V}$ tomáramos todos los *patches* de todas las fotografías, la carga computacional que resultaría sería enorme. Por eso, se ha optado por tomar de forma aleatoria un cierto número de *patches* ‘*NP*’ en cada imagen de la base, por lo que se asume que es posible que haya solapamiento entre ellos. De esta forma conseguiremos de forma aproximada cubrir la mayoría de las zonas de la imagen y haciendo uso de un menor número de datos.



Figura 6.3 En esta figura se representa una imagen de la base sobre la que se ha colocado una distribución de patches. Dichos patches se toman de forma aleatoria, pudiendo incluso existir solapamiento entre ellos. Con esto conseguimos cubrir la mayoría de zonas de la fotografía con un menor número de datos.

La matriz V se irá conformando mediante la colocación de cada uno de los patches obtenidos en forma de vector columna, de forma que al final de esta fase de preprocesado tendrá ' $num_im \times NP$ ' columnas y tam^2 filas.

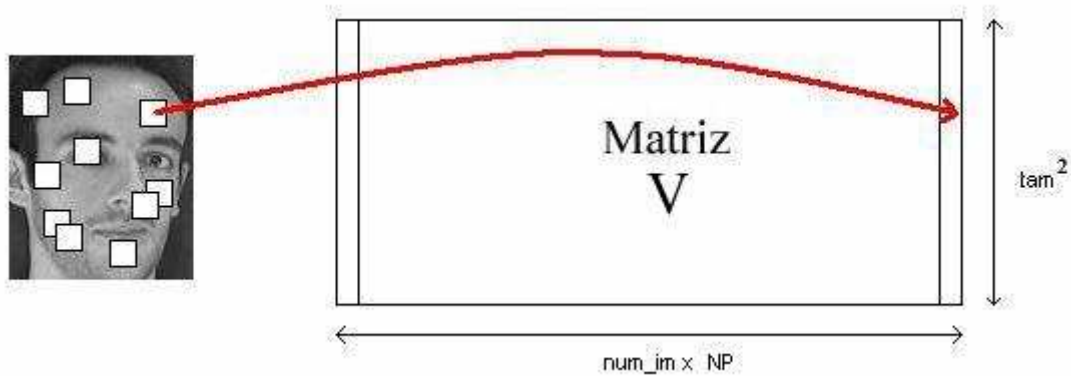


Figura 6.4 Colocación de los patches representativos de las caras de la matriz en forma de vectores columna de tamaño ' tam^2 '. En total tendremos ' $num_im \times NP$ ' columnas en V .

6.2.2.2 Preprocesado de la imagen a reconstruir

En el caso de la imagen original, el preprocesado que se le aplique viene condicionado sobremanera por la estrategia que vayamos a seguir de cara a la reconstrucción de la misma. Adelantando algunos conceptos que veremos posteriormente, en los experimentos se implementaron dos técnicas de reconstrucción, una de ellas consistía en realizar desplazamientos de la imagen original completa un cierto número de píxeles en tres direcciones del espacio (hacia la derecha, hacia abajo y hacia abajo y la derecha), para así poder hacer frente a posibles desplazamientos que se pudieran producir en la imagen original respecto a las bases. En este caso, la reconstrucción se hará patch a patch. La otra técnica se fundamenta en ir reconstruyendo la imagen original píxel a píxel por lo que no requiere realmente ningún tratamiento previo significativo.

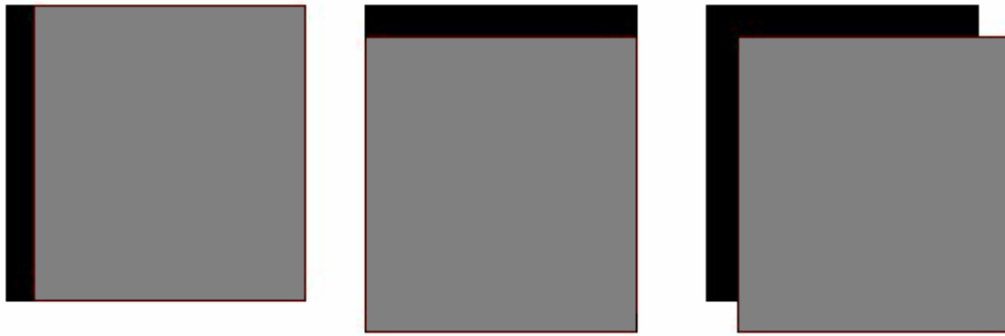


Figura 6.5 Esquema representativo del desplazamiento de la imagen original en tres direcciones del espacio para su posterior reconstrucción. En color negro aparece la imagen original y superpuestas (en color gris) las imágenes desplazadas.

Para el caso de la reconstrucción patch a patch, tendremos que ajustarnos al máximo de patches posibles que se puedan tomar ya que la fotografía se quiere reconstruir entera. Dado que al tomar patches de tamaño tam filas y columnas puede que no sea posible recorrer toda la imagen sin salirnos del soporte de la misma, posiblemente haya que descartar algunos píxeles correspondientes a los bordes. Por ejemplo, si el lado del patch es de 6 píxeles, dado que la imagen tiene un tamaño de 112 filas y 92 columnas, no será posible abarcar toda la imagen sin realizar solapamientos entre patches.

6.2.3 Aplicación del algoritmo NMF

Una vez generada la matriz de entrenamiento $\mathbf{V}$, procederemos a ejecutar el algoritmo NMF tal como se describió en el capítulo teórico correspondiente, con el objeto de obtener las bases que permitirán reconstruir la imagen original.

El algoritmo de obtención de bases requiere como parámetros tanto el número de bases a obtener, que vendrá dado por la variable ' r ' como la matriz de entrenamiento. Adicionalmente podremos indicar el número máximo de iteraciones que se llevarán a cabo. Es preciso recordar que este método *iterativo*, se fundamenta en la maximización de una función de coste, por lo que mientras mayor sea el número de ejecuciones del algoritmo, más precisos serán los resultados. Tendremos que llegar a un compromiso entre carga computacional y precisión a la hora de asignar este parámetro de entrada.

Una vez aplicado el algoritmo, dispondremos de las matrices $\mathbf{W}$ y $\mathbf{H}$, de las bases y codificación respectivamente. Realmente tan sólo $\mathbf{W}$ resultará de interés para el resto del problema, cuyas dimensiones será de ' tam^2 ' filas y ' r ' columnas. En cada una de las columnas de $\mathbf{W}$ encontraremos las bases que permitirán reconstruir la imagen original.

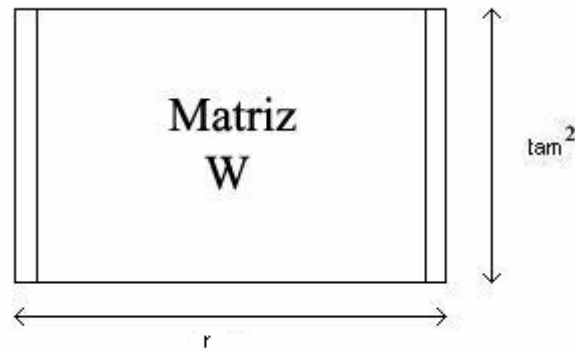


Figura 6.6 La matriz W contiene en sus columnas cada uno de los patches que representan las bases que permitirán reconstruir la imagen original.

6.2.4 Proyección sobre las bases

En este apartado veremos la obtención de los coeficientes que determinarán la proyección de cada uno de los patches que se tengan en cuenta de la imagen original sobre las bases obtenidas previamente. Desarrollaremos la idea fijándonos tan sólo en la proyección de un solo patch sobre las bases. Esto se puede extender a cada uno de los patches que conformen la imagen en función de la técnica de reconstrucción empleada.

El procedimiento de proyección consiste básicamente en ir realizando el producto escalar del vector columna de cada patch por el vector representativo de cada base. Una vez realizado dicho producto interno y normalizando posteriormente, tendremos los coeficientes de la proyección correspondientes de dicho patch sobre las bases.

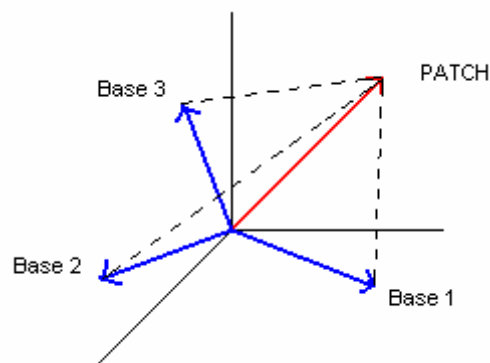


Figura 6.7 Ejemplo representativo de las proyecciones de un patch (en color rojo), sobre tres bases (en color azul). Las proyecciones se modelan como el producto escalar entre el vector del patch y los vectores de las bases.

Una vez realizadas todas las proyecciones, obtendremos una matriz que tendrá ' r ' filas (tantas como bases) y tantas columnas como patches contenga la imagen original.

6.2.5 Eliminación del ruido aditivo gaussiano

Como hemos visto en apartados anteriores, la imagen original se encuentra inmersa en un ruido aditivo y gaussiano que habrá que eliminar, es decir:

$$\mathbf{u}_{noise} = \mathbf{u} + \mathbf{n} \quad (6.1)$$

donde $\mathbf{u}_{noise}$ es la imagen ruidosa y $\mathbf{u}$ y $\mathbf{n}$, representan la imagen original y el ruido aditivo respectivamente. El ruido $\mathbf{n}$ supondremos además que es blanco, gaussiano, de media cero y varianza σ^2 .

Sin embargo, el problema de la eliminación del ruido no lo abordaremos directamente sobre la imagen, sino que lo haremos a partir de las proyecciones de cada uno de los patches sobre las bases. Dichas proyecciones, que proceden de patches que también contienen ruido serán a su vez ‘ruidosas’ en cuanto a que no representan realmente el producto escalar del vector del patch sobre la base. La presencia del ruido aditivo sobre la imagen provocará que los vectores que conforman cada uno de los patches, presenten una pequeña desviación respecto a su posición original cuando la imagen carece de ruido alguno.

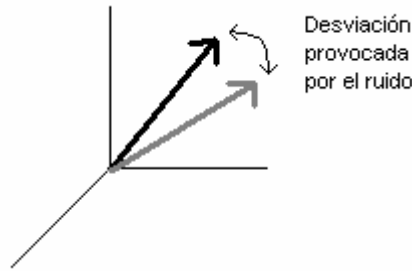


Figura 6.8 La presencia de ruido provoca que el vector representativo del patch original (en color negro) presente una ligera desviación de su posición original (en color gris). Esto provocará que las proyecciones resulten también ‘ruidosas’.

El algoritmo de eliminación de ruido pretenderá por tanto estimar las proyecciones de forma que se cancele el efecto provocado por la presencia del ruido gaussiano. A partir de $\mathbf{proy}_{i,noise}$, vector de proyecciones ruidosas correspondiente al i -ésimo patch de la imagen, se pretenderá estimar $\mathbf{proy}_i$, vector de proyecciones en el caso en el que no hay ruido:

$$\hat{\mathbf{proy}}_i = \phi(\mathbf{proy}_{i,noise}) \quad (6.2)$$

siendo ϕ la función de transformación y $\hat{\mathbf{proy}}_i$ la estimación de la proyección.

Dado que el ruido es gaussiano, será necesario comprobar que la separación del vector de proyecciones sin ruido y el ruido es viable. Para ello habrá que imponer la condición de que el vector de proyecciones no puede ser gaussiano, ya que la suma de dos variables gaussianas es a su vez gaussiana, y la separación no sería posible. En general, la distribución de probabilidad que siguen las imágenes es supergaussiana. Sin embargo, a priori no sabemos el tipo de distribución que van a seguir las proyecciones de los patches sobre las bases, aunque dado que su obtención se produce a partir del

producto de variables supergaussianas, podremos considerar que mantiene dicha distribución de probabilidad.

Sin embargo, a priori no sabemos el grado de ‘supergaussianidad’ que tiene la variable aleatoria. Generalmente se suele hacer una clasificación en torno a tres tipos de variables:

- *Muy dispersas*, si $\sqrt{E\{\mathbf{proy}_i^2\}}p(0) > \frac{1}{\sqrt{2}}$.
- *Moderadamente dispersas*, si $\sqrt{E\{\mathbf{proy}_i^2\}}p(0) < \frac{1}{\sqrt{2}}$.
- *Laplacianas*, si $\sqrt{E\{\mathbf{proy}_i^2\}}p(0) = \frac{1}{\sqrt{2}}$.

En todo caso, tendremos que hacer los cálculos pertinentes de cara a determinar de forma factible el tipo de función de distribución correcto para aplicar la técnica correcta de reducción de ruido.

Seguidamente desarrollaremos un método de estimación de *Máxima Verosimilitud (ML)* para estimar las proyecciones. La bibliografía básica en la que nos basaremos será [Jenssen00].

6.2.5.1 El método de Máxima Verosimilitud

El método de Máxima Verosimilitud se basa en la elección del vector $\mathbf{proy}_i$ que hace que $\mathbf{proy}_{i,noise}$ sea lo más probable posible. Dado que tanto $\mathbf{proy}_i$ como $\mathbf{proy}_{i,noise}$ son variables aleatorias, podemos definir su función densidad de probabilidad conjunta que denotaremos por $p_{proy}(\mathbf{proy}_i / \mathbf{proy}_{i,noise})$. El método de Máxima Verosimilitud pretende encontrar aquel valor de $\mathbf{proy}_i$ que maximiza $p_{proy}(\mathbf{proy}_i / \mathbf{proy}_{i,noise})$ para un cierto valor del vector de proyecciones ruidosas que llamaremos $\mathbf{proy}'_{i,noise}$.

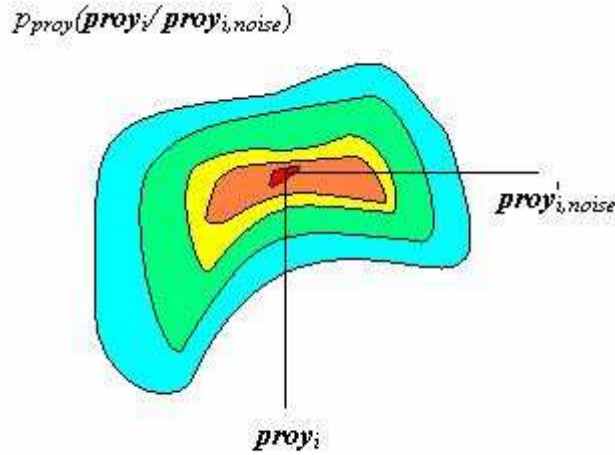


Figura 6.9 El método de Máxima Verosimilitud pretende maximizar la función densidad de probabilidad conjunta para un cierto valor $\text{proy}'_{i,\text{noise}}$ dado, es decir, se trata de elegir el valor de proy_i que maximiza $p_{\text{proy}}(\text{proy}_i / \text{proy}_{i,\text{noise}})$. En la figura se representan las curvas de nivel de una cierta función densidad de probabilidad conjunta junto con los valores de los vectores representativos de las proyecciones con y sin ruido.

Matemáticamente podemos expresarlo como:

$$\hat{\text{proy}}_i = \arg \max_{\text{proy}_i} \left\{ p(\text{proy}_i / \text{proy}'_{i,\text{noise}}) \right\} \quad (6.3)$$

Una vez definido el objetivo del estimador de Máxima Verosimilitud, el siguiente paso consistirá en determinar la función ϕ que aparece en la ecuación (6.2). Para ello, en primer lugar aplicaremos algunos principios estadísticos que permitan transformar la ecuación (6.3) para así poder formular la expresión más usual del estimador ML:

$$p_{\text{proy}}(\text{proy}_i, \text{proy}_{i,\text{noise}}) = p_{\text{proy}}(\text{proy}_i / \text{proy}_{i,\text{noise}}) p_{\text{proy}}(\text{proy}_{i,\text{noise}}) \quad (6.4)$$

que relacionada con (6.3) resultará:

$$\begin{aligned} \hat{\text{proy}}_i &= \arg \max_{\text{proy}_i} \left\{ p(\text{proy}_i / \text{proy}'_{i,\text{noise}}) \right\} = \\ &= \arg \max_{\text{proy}_i} \left\{ p(\text{proy}_i, \text{proy}'_{i,\text{noise}}) p_{\text{proy}}(\text{proy}_{i,\text{noise}}) \right\} = \\ &= \arg \max_{\text{proy}_i} \left\{ p(\text{proy}_i, \text{proy}'_{i,\text{noise}}) \right\} \end{aligned} \quad (6.5)$$

ya que $p_{\text{proy}}(\text{proy}_{i,\text{noise}})$ es independiente de proy_i , por lo que no jugará ningún papel en las estimación.

Por otro lado, es posible encontrar una formulación alternativa de la expresión (6.4) usando el Teorema de Bayes:

$$p_{\text{proy}}(\mathbf{proy}_i, \mathbf{proy}_{i,\text{noise}}) = p_{\text{proy}}(\mathbf{proy}_{i,\text{noise}}/\mathbf{proy}_i) p_{\text{proy}}(\mathbf{proy}_i) \quad (6.6)$$

de esta forma, dado que el ruido $\mathbf{n}$ es de media cero y varianza σ^2 , podremos determinar una expresión para $p_{\text{proy}}(\mathbf{proy}_{i,\text{noise}}/\mathbf{proy}_i)$:

$$\begin{aligned} p_{\text{proy}}(\mathbf{proy}_{i,\text{noise}}/\mathbf{proy}_i) &= p_{\text{proy},N}(\mathbf{proy}_{i,\text{noise}} - \mathbf{proy}_i) = \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{proy}_{i,\text{noise}} - \mathbf{proy}_i)^2}{2\sigma^2}\right) \end{aligned} \quad (6.7)$$

y por tanto, llegaremos a:

$$\begin{aligned} \hat{\mathbf{proy}}_i &= \arg \max_{\mathbf{proy}_i} \{p(\mathbf{proy}_i, \mathbf{proy}'_{i,\text{noise}})\} = \\ &= \arg \max_{\mathbf{proy}_i} \{\log p(\mathbf{proy}_i, \mathbf{proy}'_{i,\text{noise}})\} \end{aligned} \quad (6.8)$$

siendo $\log p(\mathbf{proy}_i, \mathbf{proy}'_{i,\text{noise}})$ la función *logarítmica de verosimilitud*.

Esta expresión anterior se puede transformar, de forma que el estimador resulte en función de la conocida como *Autoinformación* de la variable aleatoria $\mathbf{proy}_i$, que llamaremos $a(\mathbf{proy}_i)$:

$$\begin{aligned} \hat{\mathbf{proy}}_i &= \arg \max_{\mathbf{proy}_i} \{\log p(\mathbf{proy}_{i,\text{noise}}/\mathbf{proy}_i) p(\mathbf{proy}_i)\} = \\ &= \arg \max_{\mathbf{proy}_i} \{\log p(\mathbf{proy}_{i,\text{noise}}/\mathbf{proy}_i) + \log \mathbf{proy}_i\} = \\ &= \arg \max_{\mathbf{proy}_i} \{\log p(\mathbf{proy}_{i,\text{noise}}/\mathbf{proy}_i) - a(\mathbf{proy}_i)\} \end{aligned} \quad (6.9)$$

y aplicando las propiedades del logaritmo, podremos transformar la expresión (6.7) de la siguiente forma:

$$\begin{aligned} \hat{\mathbf{proy}}_i &= \log p(\mathbf{proy}_{i,\text{noise}}/\mathbf{proy}_i) = \log \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\mathbf{proy}_{i,\text{noise}} - \mathbf{proy}_i)^2}{2\sigma^2}\right) = \\ &= \log \frac{1}{\sigma\sqrt{2\pi}} - \frac{(\mathbf{proy}_{i,\text{noise}} - \mathbf{proy}_i)^2}{2\sigma^2} \end{aligned} \quad (6.10)$$

Dado que en la expresión anterior aparece un término constante, éste se podrá suprimir y por tanto obtener una expresión definitiva del estimador empleando la Autoinformación:

$$\begin{aligned}
\hat{\mathbf{proy}}_i &= \arg \max_{\mathbf{proy}_i} \left\{ -\frac{(\mathbf{proy}_{i,\text{noise}} - \mathbf{proy}_i)^2}{2\sigma^2} - a(\mathbf{proy}_i) \right\} = \\
&= \arg \min_{\mathbf{proy}_i} \left\{ \frac{(\mathbf{proy}_{i,\text{noise}} - \mathbf{proy}_i)^2}{2\sigma^2} + a(\mathbf{proy}_i) \right\}
\end{aligned} \tag{6.11}$$

Considerando que la Autoinformación se corresponde con una función convexa y diferenciable en todo su dominio, la obtención del estimador de Máxima Verosimilitud tan sólo se limitará a la resolución de un problema de optimización:

$$\begin{aligned}
\frac{d}{d\mathbf{proy}_i} \left\{ \frac{(\mathbf{proy}_{i,\text{noise}} - \hat{\mathbf{proy}}_i)^2}{2\sigma^2} + a(\hat{\mathbf{proy}}_i) \right\} &= 0 \\
\frac{(\hat{\mathbf{proy}}_i - \mathbf{proy}_{i,\text{noise}})^2}{\sigma^2} + \Psi(\hat{\mathbf{proy}}_i) &= 0
\end{aligned} \tag{6.12}$$

donde $\Psi(\hat{\mathbf{proy}}_i) = \frac{d}{d\mathbf{proy}_i} a(\hat{\mathbf{proy}}_i)$ es la que se conoce como *función ‘Score’* de la variable $\mathbf{proy}_i$.

Como comentamos al principio de este apartado, es necesario saber la función densidad de probabilidad de la variable $\mathbf{proy}_i$, que por ejemplo aparece en el cómputo de su Autoinformación. Consideraremos que $\mathbf{proy}_i$ sigue una distribución de tipo *laplaciana* y por tanto *supergaussiana* a efectos de implementación del método de Máxima Verosimilitud, como veremos seguidamente.

6.2.5.2 El método de Máxima Verosimilitud para variables supergaussianas

En este subapartado se van a facilitar las expresiones que permiten obtener el estimador de Máxima Verosimilitud para cada uno de los tres tipos de variables supergaussianas, es decir, *laplacianas*, *moderadamente dispersas* y *muy dispersas*. La bibliografía que emplearemos en esta sección será [Jenssen00].

1. Variables Laplacianas

Dada una variable aleatoria laplaciana $\mathbf{proy}_i$, su función densidad de probabilidad será del tipo:

$$p(\mathbf{proy}_i) = \frac{1}{\sqrt{2}\theta} \exp\left(-\frac{\sqrt{2}|\mathbf{proy}_i|}{\theta}\right) \tag{6.13}$$

siendo θ la desviación típica de $\mathbf{proy}_i$.

Para proceder a la estimación ML de $\mathbf{proy}_i$, haremos uso de la expresión (6.12), si bien previamente tendremos que determinar la función ‘Score’:

$$\begin{aligned}\Psi(\mathbf{proy}_i) &= -\frac{d}{d\mathbf{proy}_i} \log p(\mathbf{proy}_i) = \\ &= -\frac{d}{d\mathbf{proy}_i} \left\{ -\log \sqrt{2\theta} - \frac{\sqrt{2}|\mathbf{proy}_i|}{\theta} \right\} = \\ &= \frac{d}{d\mathbf{proy}_i} \left\{ \frac{\sqrt{2}|\mathbf{proy}_i|}{\theta} \right\} = \frac{\sqrt{2}}{\theta} \text{sign}(\mathbf{proy}_i)\end{aligned}\quad (6.14)$$

de forma, que insertándola en la expresión del estimador de Máxima Verosimilitud de $\mathbf{proy}_i$ dada en la expresión (6.11), llegaremos a:

$$\hat{\mathbf{proy}}_i = \phi(\mathbf{proy}_{i,noise}) = \text{sign}(\mathbf{proy}_{i,noise}) \max(0, |\mathbf{proy}_{i,noise}| - \frac{\sqrt{2}\sigma^2}{\theta}) \quad (6.15)$$

que es la que se conoce como *función de Reducción*, que implica la asignación de valores nulos a todas aquellas proyecciones cuyos valores estén comprendidos en el intervalo $\left[-\frac{\sqrt{2}\sigma^2}{\theta}, \frac{\sqrt{2}\sigma^2}{\theta} \right]$.

2. Variables moderadamente dispersas

Realizando un razonamiento análogo al planteado en el apartado referente a las variables laplacianas, llegaremos al estimador ML deseado, que será:

$$(6.16)$$

donde las constantes a y b vienen determinadas por las siguientes expresiones:

$$b = \frac{2p(0)E\{\mathbf{proy}_i^2\} - E\{|\mathbf{proy}_i|\}}{E\{\mathbf{proy}_i^2\} - (E\{|\mathbf{proy}_i|\})^2} \quad (6.17)$$

$$a = \frac{1}{E\{\mathbf{proy}_i^2\}} (1 - E\{|\mathbf{proy}_i|\}b) \quad (6.18)$$

siendo $p(0)$ el valor que toma la función densidad de probabilidad en el origen.

3. Variables muy dispersas

Procediendo de forma análoga obtendremos que la *función de Reducción* para el caso de variables muy dispersas tendrá la siguiente expresión:

$$\begin{aligned}\hat{\mathbf{proy}}_i &= \phi(\mathbf{proy}_{i,noise}) = \\ &= \text{sign}(\mathbf{proy}_{i,noise}) \max \left(0, \frac{|\mathbf{proy}_{i,noise}| - a\theta}{2} + \frac{1}{2} \sqrt{(|\mathbf{proy}_{i,noise}| - a\theta)^2 - 4\sigma^2(\alpha + 3)} \right) \end{aligned} \quad (6.19)$$

siendo $a = \sqrt{0.5\alpha(\alpha + 1)}$ donde:

$$\alpha = \frac{2 - k + \sqrt{k(k + 4)}}{2k - 1} \quad (6.20)$$

con $k = (\theta p(0))^2$.

Además contamos con una expresión cerrada que permite obtener el valor de la densidad de probabilidad en el origen:

$$p(0) = \frac{\alpha + 2}{2\theta} [0.5\alpha(\alpha + 1)]^{-1/2} \quad (6.21)$$

En este caso, disponemos de una fórmula cuyo cómputo permite obtener de forma sencilla el valor de $p(0)$, si bien en la mayoría de las ocasiones este cálculo puede resultar bastante complicado. Para poder determinar esta cantidad dentro de la función implicada en la eliminación del ruido, se ha optado por hacer uso de la función *ksdensity* integrada dentro del *toolbox* destinado a operaciones estadísticas que proporciona Matlab. Suponiendo que la distribución tiene su máximo en el origen (si no es así, se pueden realizar los cambios oportunos para que lo sea), tomaremos aquel valor del vector devuelto por *ksdensity* que sea mayor y lo consideraremos como $p(0)$.

El fundamento de esta función está relacionado con el *Método de Enventanado de Parzen* que proporciona una expresión aproximada de la función densidad de probabilidad de un cierto conjunto de muestras de una variable aleatoria. Dada una variable aleatoria a su función densidad estimada será:

$$\hat{p}_a(a) = \frac{1}{nh} \sum_{k=1}^n K\left(\frac{a - a_k}{h}\right) \quad (6.22)$$

donde a_k con $k = 1, \dots, n$ son realizaciones de a y además:

$$\int_{-\infty}^{\infty} K dx = 1 \quad (6.23)$$

siendo K una densidad gaussiana estándar cuya anchura viene determinada por el parámetro h .

6.2.6 Reconstrucción de las imágenes

Una vez se ha realizado el proceso de eliminación de ruido en las proyecciones de los patches sobre las bases, se va a proceder a la reconstrucción de la imagen original. Como se comentó en apartados previos, el proceso de reconstrucción se realizó de dos formas diferentes en función de si se hacía patch a patch o bien píxel a píxel. En este apartado estudiaremos las dos estrategias desarrolladas, si bien veremos como sólo en el segundo caso se obtendrán resultados satisfactorios, por lo que finalmente será la técnica empleada.

6.2.6.1 Reconstrucción patch a patch

Como se adelantó previamente, este método se fundamenta en el desplazamiento de la imagen original en una serie de direcciones del espacio de cara a solventar los posibles problemas que se podrían presentar en cuanto a ligeras variaciones de la posición de las bocas, ojos, etc entre la imagen original y las bases.

Dada una de estas imágenes desplazadas, el método de reconstrucción patch a patch comienza con la división en un cierto número de patches cuadrados de tamaño 'tam'. Seguidamente, a cada uno de los patches se le aplica la técnica descrita en el apartado anterior referente a la proyección sobre las bases y eliminación de ruido. Tras ello, cada uno de los patches en los que se dividió la imagen original se reconstruye como la suma de las bases ponderada por los coeficientes de proyección. Los patches estimados serán colocados en la posición que les correspondía en la imagen inicial.

En la siguiente figura se presenta un esquema que explica la síntesis de una de las tres imágenes obtenidas tras los desplazamientos.

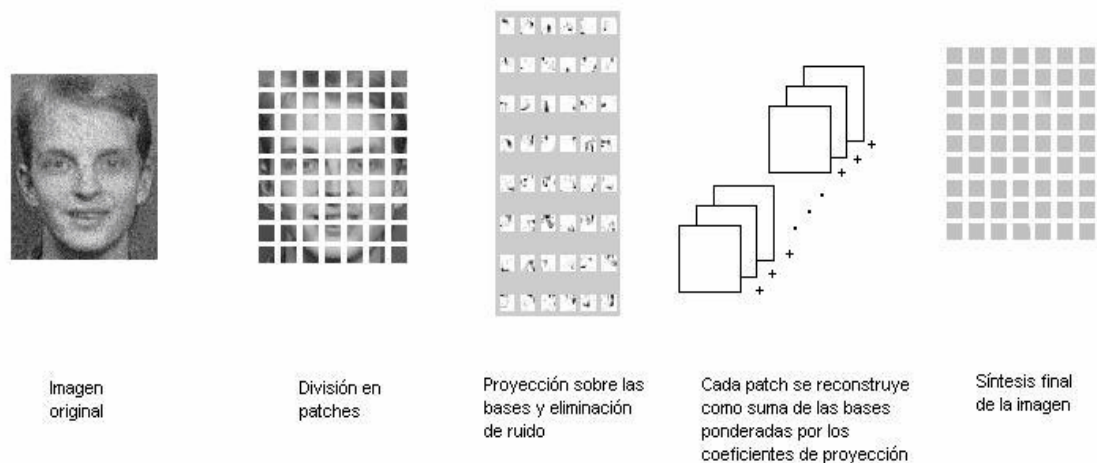


Figura 6.10 Esquema representativo del proceso de reconstrucción patch a patch de una de las imágenes procedentes del desplazamiento en una de las direcciones del espacio. El resultado final será el promediado de cada una de estas imágenes.

Por último, la obtención de la imagen definitiva se hará promediando todas las imágenes obtenidas tras los desplazamientos. Si originalmente se obtuvieron R imágenes a partir de los desplazamientos I_j , la imagen final I_{final} será:

$$I_{final} = \frac{1}{R} \sum_j I_j \quad (6.24)$$

donde la suma se ha realizado píxel a píxel entre todas ellas.

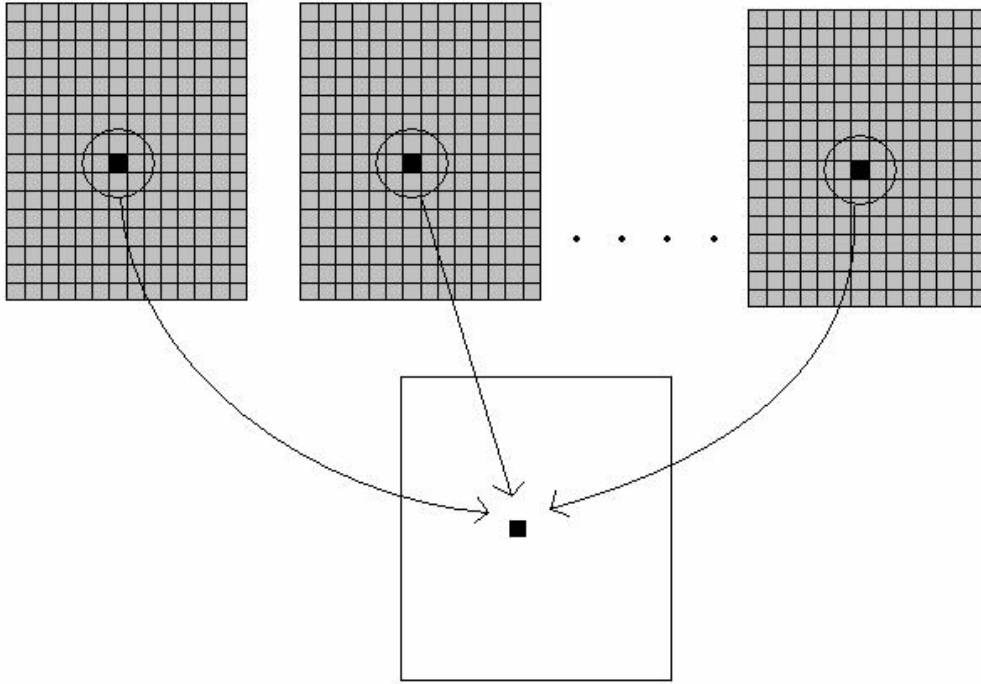


Figura 6.11 Las imagen final se reconstruye promediando píxel a píxel las imágenes estimadas que fueron desplazadas inicialmente.

6.2.6.2 Reconstrucción píxel a píxel

Esta técnica de reconstrucción, a diferencia de la anterior, se basa en ir estimando el valor de un píxel en cada iteración y no un patch completo. Por ello, la estimación será más precisa que en el caso estudiado previamente. Dado un cierto píxel, que denotaremos por su fila ' i ' y columna ' j ' como P_{ij} , para la reconstrucción del mismo utilizaremos la información de un cierto número de píxeles adyacentes a determinar.

Los píxeles de la vecindad de P_{ij} que usaremos serán aquellos que se obtengan de la generación de un cierto número de patches NP que contendrán al píxel P_{ij} y que tendrán tamaño ' tam '. Para fijar ideas, supondremos el caso en el que NP vale cuatro y el tamaño del patch es de dos píxeles y lo representaremos gráficamente con el objeto de ilustrar el concepto:

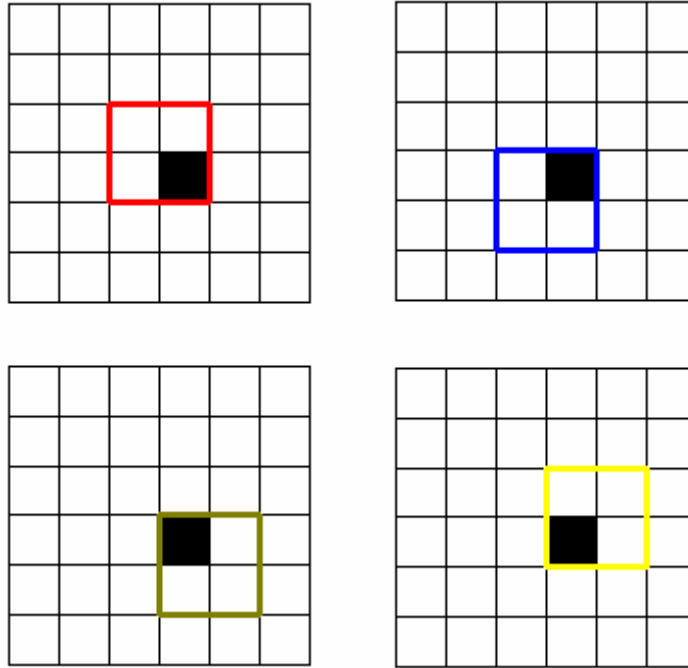


Figura 6.12 Generación de patches en torno a un cierto píxel P_{ij} (en color negro en la figura), conteniendo los píxeles adyacentes. En este caso, el número de patches es cuatro y el tamaño del patch es de dos píxeles.

Como se puede deducir fácilmente, para un tamaño de patch ' tam ' dado, el número máximo de patches que se pueden tomar que contengan a un mismo píxel (y sin que haya repetición), es de $NP = tam^2$. Este aspecto habrá que tenerlo en cuenta a la hora de ver con cuántos patches hemos de trabajar sin que ello implique un exceso de información por píxel, como veremos en el apartado referente a las simulaciones realizadas.

Una vez tenemos los NP patches se seguirá el método habitual consistente en la proyección sobre las bases y eliminación del ruido como se vio previamente. Sin embargo, una vez llegados al punto en el que se desea reconstruir el píxel, hay que observar que en cada patch que hemos reconstruido, el píxel ocupa una posición distinta debido a la forma en la que se han tomado. Por ello hay que tener perfectamente ubicado el píxel a reconstruir en cada patch para sumar la posición correcta.

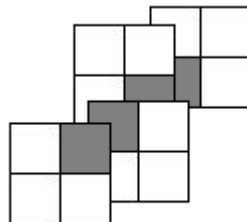


Figura 6.13 Representación tridimensional del proceso de ajuste que permite sumar las aportaciones de cada patch en el píxel que se está reconstruyendo.

6.2.7 Evaluación de resultados

Si bien a partir de valoraciones a simple vista podemos tener una idea aproximada del correcto funcionamiento, hemos de contar con un criterio objetivo que permita ver si los experimentos realizados son satisfactorios, a la vez que muestre la influencia positiva o negativa de la variación de los parámetros de la simulación. En nuestro caso, contaremos con la *Relación Señal a Ruido (SNR)*.

Por el contrario, la SNR para la imagen estimada, que denotaremos por SNR' , tendrá una expresión, que si bien no es del todo rigurosa si nos permitirá obtener una visión bastante aceptable de la reducción de ruido resultante en la imagen reconstruida:

$$SNR' = 20 \log \left(\frac{\sigma_{original}}{\sigma_{(original-estimada)}} \right) \quad (6.25)$$

donde $\sigma_{(original-estimada)}$ es la desviación estándar de la imagen diferencia entre la original y la estimada.

Una forma alternativa de calcular esta expresión consiste en estimar la varianza del ruido como la suma (píxel a píxel) de los cuadrados de las diferencias entre las imágenes original y estimada, de forma que obtendríamos la siguiente expresión equivalente:

$$SNR' = 10 \log \left(\frac{\sum_{i,j} P_{ij,original}^2}{\sum_{i,j} (P_{ij,original} - P_{ij,estimada})^2} \right) \quad (6.26)$$

Como veremos en el apartado siguiente, las imágenes reconstruidas presentan un cierto efecto de emborronamiento provocado por los diferentes procesos de promediados que se realizan en el algoritmo de reconstrucción. Este efecto se tendrá en cuenta como ruido ya que la diferencia entre los píxeles de la imagen original y estimada va a ser mayor, por lo que la SNR' calculada disminuirá considerablemente. Por tanto esta medida de la relación señal a ruido tan sólo se ha de tener en cuenta con el fin de comparar entre diferentes medidas realizadas y ver cuál se comporta mejor frente al ruido.

6.3 Experimentos

En este apartado estudiaremos las diferentes simulaciones realizadas para la reconstrucción de imágenes ruidosas variando alguno de los parámetros del programa de cara a observar su influencia en los resultados obtenidos. En primer lugar se verán las técnicas basadas en la *Factorización No Negativa de Matrices* que se han desarrollado previamente y posteriormente haremos uso del algoritmo FastICA y PCA para comparar los resultados.

6.3.1 Resultados obtenidos mediante técnicas NMF

Seguidamente se presentarán los resultados obtenidos tras la aplicación de la *Factorización No Negativa de Matrices (NMF)*, para la obtención de las bases usadas para la reconstrucción de las imágenes ruidosas. Como se indicó en apartados anteriores, se han empleado dos técnicas para sintetizar las imágenes originales a partir de las bases, que son la reconstrucción patch a patch y píxel a píxel. Veremos como la primera de ellas ofrece resultados que no serán satisfactorios, por lo que finalmente nos inclinaremos por la segunda opción de cara a continuar con el resto de simulaciones.

6.3.1.1 Reconstrucción patch a patch

Recordando la idea fundamental de esta técnica, veremos que basa en reconstruir de forma individual un conjunto de imágenes procedentes de la imagen original, que han sido desplazadas un cierto número de píxeles en varias direcciones del espacio. Una vez tenemos las imágenes sintetizadas, procedemos a realizar un promediado entre todas de forma que lugar a la imagen final.

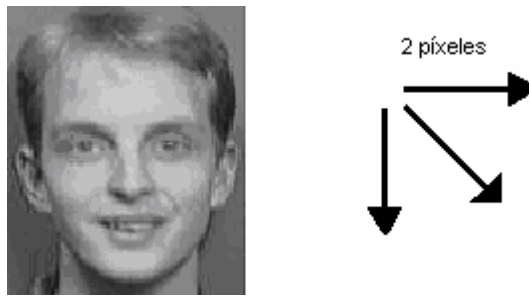


Figura 6.14 *Imagen original a reconstruir a partir de las bases. En este primer experimento no se ha tenido en cuenta la influencia del ruido aditivo. Además se muestran las tres direcciones del espacio en las que se han realizado los desplazamientos.*

En la simulación que presentamos a continuación se han realizado desplazamientos en tres direcciones del espacio (derecha, abajo y hacia abajo y la derecha) de dos píxeles. El patch es cuadrado de lado seis píxeles y se han empleado 80 bases para generar las imágenes. Como podemos observar en la *figura 6.15* aparecen ciertos efectos que generan una especie de rejilla en el conjunto de la imagen que hace que la reconstrucción no sea correcta.

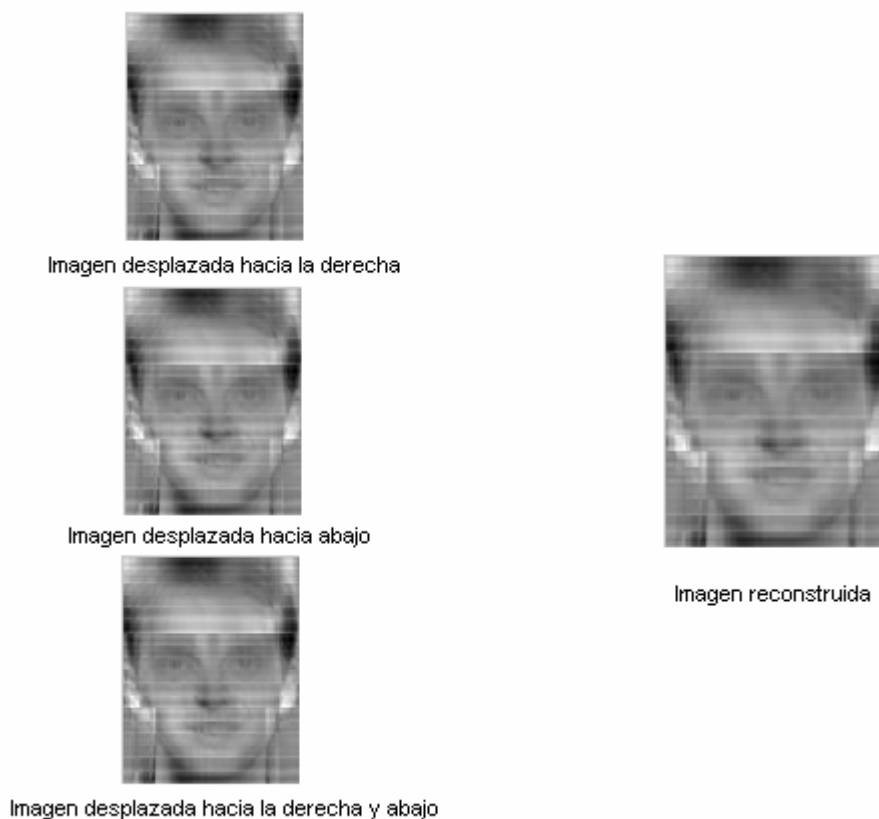


Figura 6.15 *Imágenes procedentes de la reconstrucción en las que se observa el efecto de rejilla. Aparecen tanto las imágenes sintetizadas a partir de las procedentes de los desplazamientos tanto como la imagen final resultante del promediado de las anteriores.*

Concretamente, la rejilla que se genera tiene una periodicidad que coincide con el tamaño del patch, como se puede comprobar al repetir los experimentos con diferentes valores de ‘tam’. En este caso, debido a la configuración de los desplazamientos, este efecto exclusivamente se da en la dirección horizontal. Sin embargo, si se realizan los movimientos en cuatro direcciones tales como hacia arriba, abajo, izquierda y derecha, la rejilla aparecerá tanto en vertical como en horizontal y siempre con una periodicidad coincidente con el tamaño del patch.

Por todo lo expuesto, nos vemos obligados a desechar esta técnica de reconstrucción e intentar encontrar un método alternativo en el que la presencia de los patches no se manifieste en efectos no deseados como los anteriores. Este método se basará en la reconstrucción píxel a píxel como veremos en el siguiente apartado.

6.3.1.2 Reconstrucción píxel a píxel

En esta técnica, la síntesis se realiza a partir de los píxeles tomados uno a uno, de forma que en principio no debería aparecer el efecto que aparece en la reconstrucción patch a patch y que nos llevó a desestimarla, por lo que éste será el método elegido y con el que se trabajará de aquí en adelante.

Seguidamente se van a presentar una serie de experimentos mediante los cuales se mostrarán las principales ventajas e inconvenientes de esta técnica, analizando de forma crítica la influencia que tienen los parámetros que intervienen en las simulaciones. Estas pruebas comenzarán por los casos más sencillos e irán aumentando progresivamente de complejidad para ver hasta que punto este algoritmo puede realizar las tareas para las que se diseñó.

Experimento 1: Reconstrucción de la imagen en ausencia de ruido

En primer lugar desarrollaremos la tarea más básica, que consiste en reconstruir la imagen original cuando no existe ruido aditivo. Para este experimento, el tamaño del patch es de seis píxeles de lado y se ha empleado un número de bases lo suficientemente elevado como para que la reconstrucción se pueda llevar a cabo con garantías. En la siguientes figuras se muestran los resultados obtenidos, tanto las bases, como la imagen original y la reconstruida.



Figura 6.16 Conjunto de 80 bases sobre las que se ha realizado la reconstrucción de la imagen original.

En esta simulación se han tomado 80 bases, lo cual constituye una cantidad bastante elevada en relación al número con el que se puede realizar la reconstrucción de forma satisfactoria. Podemos observar como en general no somos capaces de identificar en cada una de las bases la parte del rostro con la que se corresponde, por lo que en teoría estaremos reconstruyendo diferentes zonas de la imagen original a partir de otras que no guardan ninguna relación (por ejemplo, podremos estar reconstruyendo un ojo y

que en las bases no aparezca ninguna parte relacionada con el ojo de algunos de los rostros de entrenamiento).

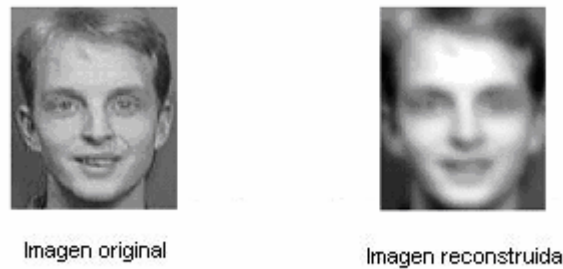


Figura 6.17 *Imágenes original y reconstruida.*

Como podemos observar en la imagen final, la reconstrucción es bastante satisfactoria y sobre todo observamos que no aparece en efecto de rejilla característico de la reconstrucción patch a patch. También hay que destacar la presencia de un cierto efecto de emborronamiento en la imagen reconstruida que vendrá dado por los diferentes procesos de promediado intermedios que hacen las veces de filtro paso de baja.

Experimento 2: Reconstrucción de la imagen en presencia de ruido

Seguidamente vamos a cuantificar los efectos del ruido aditivo en el proceso de reconstrucción de la imagen. Trabajaremos con tres valores diferentes de σ que servirán para calibrar en qué medida nuestro algoritmo es robusto frente al ruido gaussiano. Consideraremos que el número de bases es, al igual que en ejemplos anteriores, igual a 80 y el patch es un cuadrado de lado seis píxeles.

En los experimentos realizados, iremos aumentando gradualmente la potencia de ruido para ver como llegados a un cierto punto, los efectos del ruido hacen que la imagen sea prácticamente imposible de reconstruir.

Los valores empleados para las desviación típica del ruido serán $\sigma = 0.01$, $\sigma = 0.05$, y $\sigma = 0.5$, mientras que para la imagen será de $\theta = 0.3608$.

a) $\sigma = 0.01$

En este primer caso se puede observar que el ruido en el que está inmersa la imagen es prácticamente nulo, por lo que la imagen reconstruida es bastante parecida a la original (salvo el ya indicado efecto de emborronamiento). Por otro lado, hay que hacer mención a la presencia de pequeños patches oscuros que aparecen en la imagen y que son debidos al proceso de eliminación de ruido (recordemos que en la estimación de máxima verosimilitud se realizaba una umbralización, de forma que todos aquellos píxeles cuyo valor quedara por debajo de un cierto umbral, eran puestos a cero).



Figura 6.18 *Imagen ruidosa (izquierda) y reconstruida (derecha) para $\sigma = 0.01$. Observamos como la reconstrucción ha resultado bastante satisfactoria. Podemos notar la presencia de una zona oscura con forma rectangular que es debida al algoritmo de eliminación de ruido.*

El cálculo de la relación señal a ruido para esta imagen arroja un valor de 13.7816 dB. Como comentamos previamente este valor es sólo orientativo ya que el efecto de emborronamiento también se computará como ruido en la expresión de SNR' basada en el error cuadrático medio entre los píxeles.

b) $\sigma = 0.05$

Si realizamos un pequeño incremento en la potencia de ruido observamos como el ruido ya tiene mayor presencia, si bien vemos como aún no es lo suficientemente importante como para afectar a la reconstrucción final. Observamos como curiosamente vuelve a aparecer una zona oscura colocada en el mismo lugar que en el caso anterior. Como se verá posteriormente cuando se empleen distintas bases, veremos que no aparecen regiones oscuras en esa zona, por lo que a priori podemos considerar que este efecto viene dado porque el conjunto de bases que se usa en este ejemplo concreto no es capaz de reconstruir de forma adecuada esa zona.



Figura 6.19 *Imagen ruidosa (izquierda) y reconstruida (derecha) para $\sigma = 0.05$. Observamos como la reconstrucción ha resultado también bastante satisfactoria.*

En este caso, la relación señal a ruido estimada es de 12.3757 dB por lo que tan sólo notamos una ligera disminución que apenas será apreciable a simple vista.

c) $\sigma = 0.5$

En la *figura 6.20 (izquierda)* observamos como ya la potencia del ruido gaussiano es tal que resulta bastante complicado distinguir la imagen original, lo cual provocará que la reconstrucción, aunque permita intuir la forma del rostro, resulte muy defectuosa.

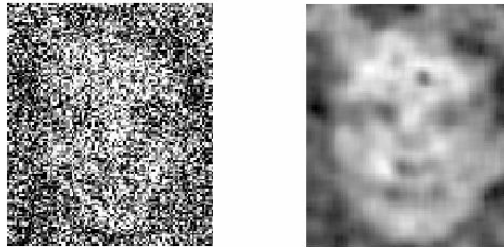


Figura 6.20 *Imagen ruidosa (izquierda) y reconstruida (derecha) para $\sigma = 0.5$. Observamos como la reconstrucción no arroja resultados correctos ya que la potencia de ruido es demasiado elevada.*

¿Cuál es el valor umbral para la desviación típica del ruido? Realmente no existe un método exacto que nos permita decidimos por uno u otro, sino que nos debemos basar en criterios puramente subjetivos. Se propone $\sigma = 0.25$ (valor que se sitúa en las proximidades de la desviación típica de la imagen original) como el umbral a partir del cual dejan de apreciarse con claridad ciertos rasgos de la imagen como los ojos o la boca.

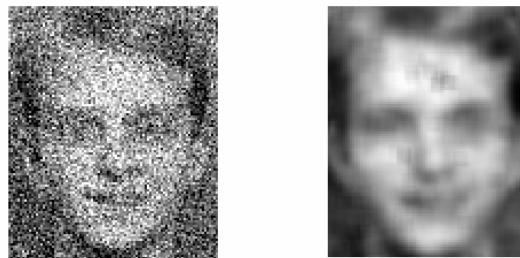


Figura 6.21 *Imagen ruidosa (izquierda) y reconstruida (derecha.) Podremos considerar que a partir de $\sigma = 0.25$, la reconstrucción deja de ser aceptable.*

Experimento 3: Reducción del número de bases y número de patches por píxel

¿Cuántas bases hemos de emplear para que la reconstrucción se encuentre dentro de los criterios requeridos? El empleo de un número de bases excesivamente elevado provocará que la carga computacional se dispare, ya que la cantidad de proyecciones a realizar aumentará considerablemente. Por tanto, interesa saber a partir de cuántas bases somos capaces de reconstruir las imágenes con garantías. En primera aproximación, convendría reducir el número de bases a un número sensiblemente menor que el que se ha venido empleando hasta el momento (ochenta bases), ya que de lo contrario el problema se estaría resolviendo con más información de la necesaria.

Veamos esta idea con mayor profundidad. Según el modelo planteado, la reconstrucción de cada píxel se realiza a partir del promediado de NP patches. Además, cada uno de estos NP patches se obtiene como resultado del proceso de reconstrucción

de la información contenida en el número de bases empleadas (que representamos por la variable ' r '). Esto implica que por cada píxel estaríamos empleando la información $NP \times rango$ patches, lo cual sin duda, en algunos casos, puede representar una cantidad excesiva de información a procesar por cada uno de los píxeles de la imagen.

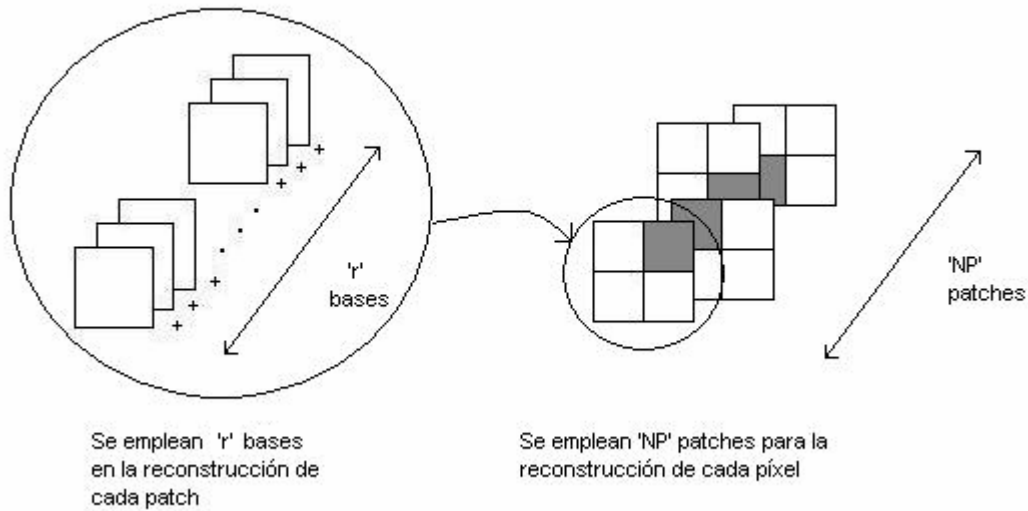


Figura 6.22 Esquema representativo del número de bases empleado en la reconstrucción de un píxel. Dado que cada píxel se reconstruye a partir de promediado de ' NP ' patches y a su vez cada uno de ellos se genera conforme a una base de ' r ' elementos, se concluye que para cada píxel se empleará la información contenida en ' $NP \times r$ ' patches.

Hasta ahora hemos venido empleando $NP = tam^2$ patches para reconstruir cada píxel, pero al igual que ocurre con las bases, la complejidad del problema resulta bastante elevada, por lo que deberíamos comprobar si realmente es posible realizar la reconstrucción tomando un número menor de patches por píxel. Si es así, deberíamos estimar de forma aproximada cuántos patches por píxel hemos de emplear para que la reconstrucción proporcione unos resultados aceptables. Lógicamente, la solución adoptada provendrá de la adopción de una solución de compromiso entre calidad de la imagen reconstruida y tiempo de procesamiento total.

Además, el empleo de un excesivo número de patches por píxel dará lugar a que el promediado realizado provoque un mayor emborronamiento de la imagen reconstruida, de forma que ciertos bordes de la imagen no aparezcan con el realce deseado.

En los siguientes experimentos se obtuvieron las imágenes sintetizadas para diferentes valores de la variable ' r ', que como se estudió previamente, representa el número de bases. Consideraremos que la imagen original está inmersa en un ruido gaussiano con $\sigma = 0.01$ y que el tamaño del patch continúa siendo un cuadrado de lado seis píxeles. Iremos disminuyendo progresivamente el número de bases empleadas: 80, 36, 18 y 9 bases y repetiremos las simulaciones para $NP = 9$, $NP = 4$ y $NP = 1$.

Una forma de cuantificar cómo de bien se ha realizado la reconstrucción consiste en emplear la Relación Señal a Ruido, tal como se definió en el apartado 6.2.7. Los resultados correspondientes a las simulaciones realizadas se presentan en la siguiente tabla.

<i>Relación Señal a Ruido (dB)</i>				
	Número de bases			
NP	9	18	36	80
9	16.2450	16.7321	17.9891	14.1237
4	16.8359	15.3037	17.1584	16.9798
1	13.5866	14.1577	14.2106	14.2177

Tabla 6.1 *Relación Señal a Ruido (en dB) de las imágenes reconstruidas para diferentes valores de ‘NP’ y ‘r’.*

Las imágenes obtenidas tras el proceso de reconstrucción se muestran en la siguiente figura:

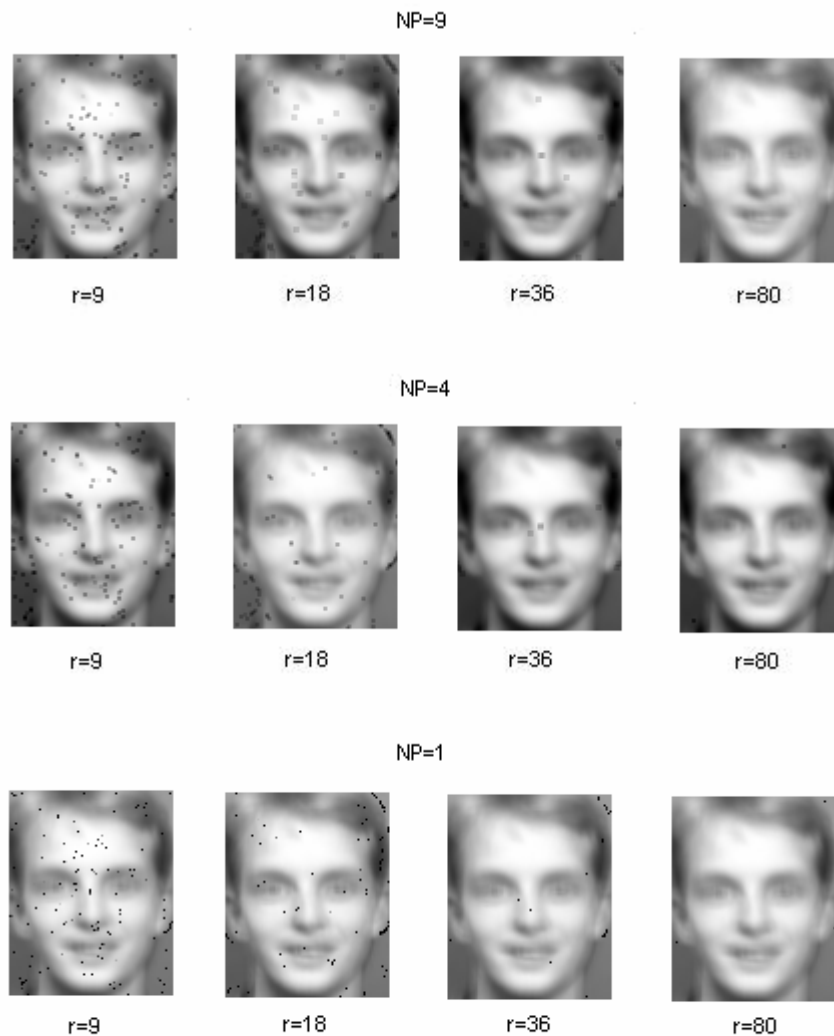


Figura 6.23 *Imágenes reconstruidas para diferentes valores de ‘NP’ y ‘r’.*

Como se puede comprobar en la *figura 6.23*, a medida que se reduce el número de bases empleadas en la reconstrucción, aparece un ruido tipo ‘sal y pimienta’ que será fruto de una eliminación de ruido basada en reducción del código disperso defectuosa. Por otro lado, la *tabla 6.1* muestra como al reducir el número de patches empleados en la reconstrucción de cada píxel, la calidad de la imagen (su SNR) disminuye progresivamente, encontrando para $NP=1$ el descenso más acusado.

6.3.2 Resultados obtenidos mediante PCA e ICA

¿Cómo sería la reconstrucción en el caso de emplear las bases obtenidas tras aplicar PCA o ICA al conjunto de datos de entrenamiento? En los siguientes experimentos daremos respuesta a esta pregunta.

Experimento 1. Extracción de bases PCA

Sirviéndonos de la función ‘pcamat.m’ de Matlab (incluida en el CD adjunto a esta memoria), extrajimos el conjunto de 36 bases PCA de tamaño 6x6 píxeles, que se muestran en la siguiente figura.

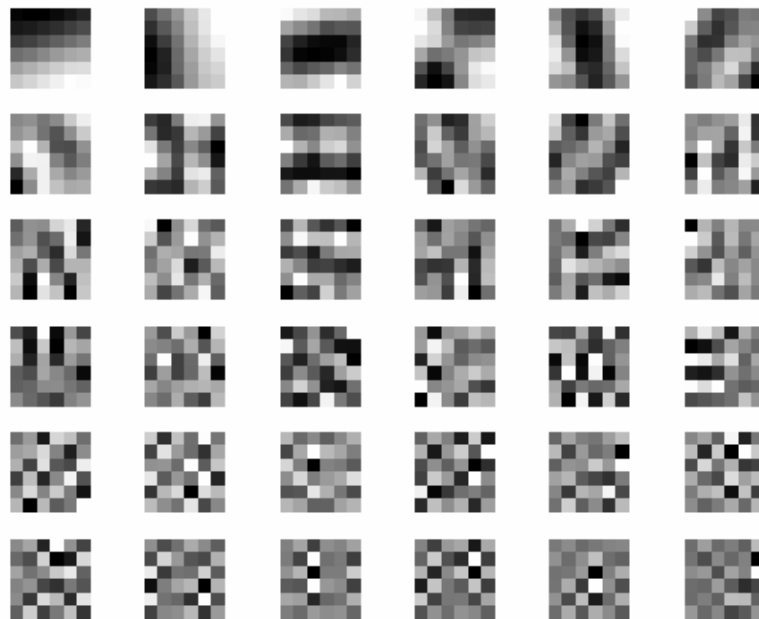


Figura 6.24 Conjunto de 36 bases extraídas correspondientes al Análisis de Componentes Principales.

Como suele ser habitual en el caso del Análisis de Componentes Principales, aparece una base asociada al autovalor dominante (esquina superior izquierda de la figura anterior) en la que se puede observar una mayor correlación entre píxeles adyacentes que en el resto de patches que conforman la base.

Experimento 2. Extracción de bases ICA

La determinación de las bases correspondientes al Análisis de Componentes Independientes se ha llevado a cabo mediante la función ‘fastica.m’ que también se proporciona en el CD adjunto. En la siguiente figura se muestran las 36 bases de tamaño 6x6 píxeles.

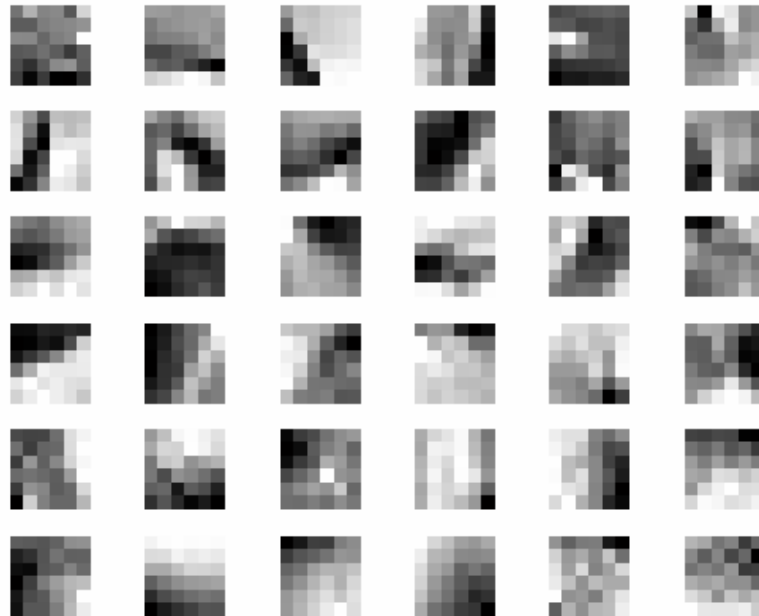


Figura 6.25 Conjunto de 36 bases extraídas correspondientes al Análisis de Componentes Independientes.

En el caso del Análisis de Componentes Independientes, la característica esencial que diferencia los resultados obtenidos de los correspondientes a otro tipo de análisis, consiste en la identificación de líneas en diferentes direcciones del espacio. En principio, se puede considerar que estos trazos que conforman las bases pretenden mostrar los diferentes bordes que aparecen en los rostros de entrenamiento.

Experimento 3. Reconstrucción de rostros mediante técnicas PCA e ICA

Una vez hemos determinado las bases correspondientes a PCA e ICA, procederemos a reconstruir uno de los rostros mediante dichas técnicas, y así poder comparar los resultados con los que proporcionaría NMF. Para ello, emplearemos un procedimiento análogo al descrito en el apartado 6.2.6.2 relativo a la reconstrucción píxel a píxel.

Como primera simulación, consideraremos la reconstrucción del rostro mediante patches de tamaño 6x6, con un solapamiento de 5 píxeles entre ellos (se correspondería realmente con el caso de la reconstrucción píxel a píxel). Los resultados obtenidos para cada una de las tres técnicas se muestran en la siguiente figura:

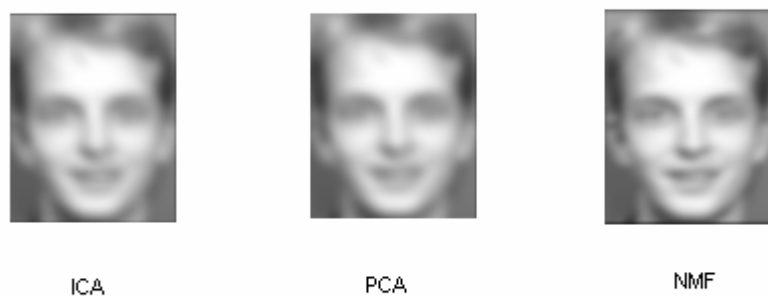


Figura 6.26 *Reconstrucción del rostro mediante ICA, PCA y NMF usando patches de tamaño 6x6 y solapamiento de 5 píxeles.*

Como podemos comprobar, la reconstrucción basada en la Factorización No Negativa de Matrices da lugar a una imagen más nítida y que permite apreciar mejor las zonas como los ojos y la boca, que presentan más bordes.

En la segunda de las simulaciones de este apartado, procederemos a reducir a 4 píxeles el solapamiento entre los patches empleados en la reconstrucción.

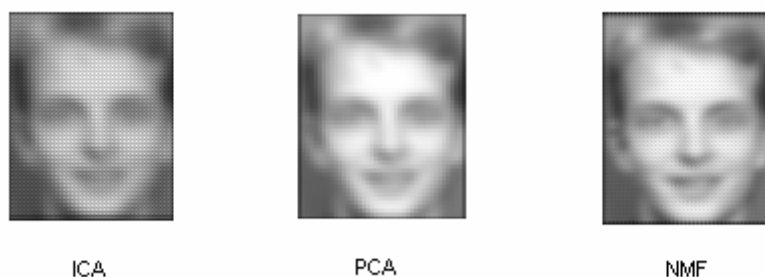


Figura 6.27 *Reconstrucción del rostro mediante ICA, PCA y NMF usando patches de tamaño 6x6 y solapamiento de 4 píxeles.*

El efecto de la disminución del solapamiento en un píxel, apenas es relevante en el caso de las imágenes sintetizadas a partir de técnicas PCA y NMF, si bien para el caso de ICA ha provocado una pérdida de calidad considerable en la reconstrucción.

Por último, volveremos a reducir el solapamiento entre patches y fijaremos su valor a 3. En este caso, notaremos como ya se comienzan a notar los efectos sobre la imagen reconstruida mediante NMF.

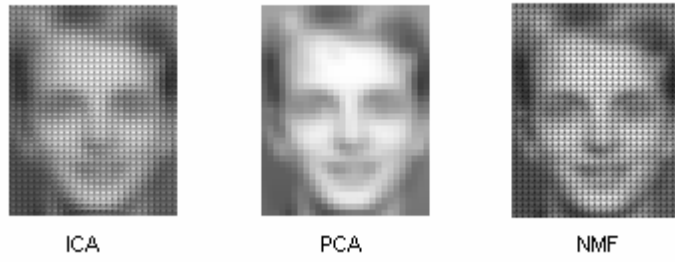


Figura 6.28 Reconstrucción del rostro mediante ICA, PCA y NMF usando patches de tamaño 6x6 y solapamiento de 3 píxeles.

A la vista de los resultados obtenidos, concluiremos que la reconstrucción basada en la Factorización No Negativa de Matrices proporciona unos resultados óptimos cuando el solapamiento entre los patches es grande. Por otro lado, el Análisis de Componentes Principales será la técnica más adecuada cuando se quieran sintetizar imágenes empleando un menor número de patches (es decir, reduciendo el solapamiento).

6.3.3 Reconocimiento facial mediante técnicas NMF

Como conclusión a este capítulo dedicado a la reconstrucción de imágenes ruidosas, comprobaremos hasta qué punto las síntesis de los rostros han sido lo suficientemente buenas como para que puedan ser identificadas con la imagen original de la cual provenían. Para ello realizaremos diferentes experimentos con varias imágenes procedentes de simulaciones anteriores, que se tratarán de clasificar de forma automática siguiendo un criterio de *mínima distancia*.

6.3.3.1 El clasificador de mínima distancia

Para poder determinar a qué cara original hay que asignar la imagen reconstruida, es necesario establecer una cierta regla de decisión. En nuestro caso se tratará de una regla de mínima distancia, que se formulará matemáticamente a partir de la *norma* de la diferencia entre cada par de imágenes conformado por un rostro de la base y la imagen a clasificar. Sean $\mathbf{v}_i$ con $i = 1, \dots, \text{num_im}$, cada una de las imágenes de la base y $\mathbf{u}$ la imagen a clasificar, entonces el *clasificador de mínima distancia* se define de la siguiente manera:

$$\min_i \{\|\mathbf{u} - \mathbf{v}_i\|\} = \min_i \left\{ \left[(\mathbf{u} - \mathbf{v}_i)^T (\mathbf{u} - \mathbf{v}_i) \right]^{1/2} \right\} \quad (6.27)$$

6.3.3.2 Experimentos

En los experimentos desarrollados intentaremos ver la influencia del ruido de la imagen original en la clasificación. Para ello, se intentarán reconocer dos imágenes, una libre de perturbaciones y otra que procede de la reconstrucción de una imagen inmersa en un ruido aditivo

Experimento 1. Clasificación de imagen no ruidosa

Consideremos una imagen no ruidosa que ha sido sintetizada mediante las técnicas descritas en los apartados anteriores. Tras aplicar el algoritmo de clasificación basado en la mínima distancia, se extrajeron las cinco imágenes más parecidas a la original.

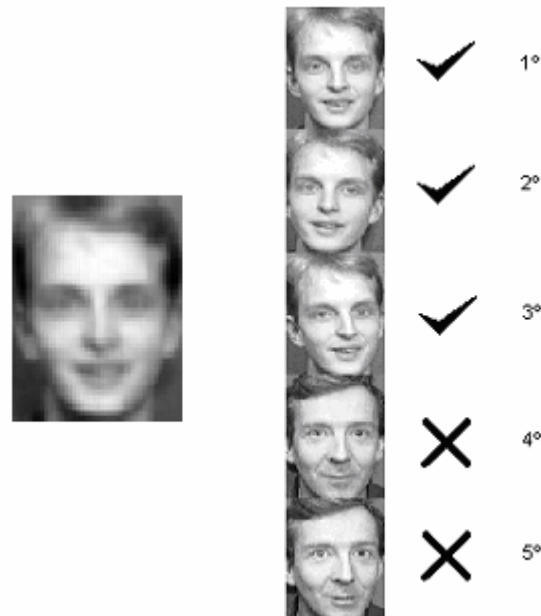


Figura 6.29 Estimación de los cinco rostros más parecidos (ordenados del primero al quinto en función de la distancia) a la imagen reconstruida (a la izquierda) atendiendo al criterio de mínima distancia. Las imágenes de la base aparecen ordenadas de la primera (la más parecida) a la quinta, y junto a ellas se representa un símbolo (✓) si la identificación fue correcta o (X) en caso contrario.

Podemos comprobar a partir de la figura anterior como el clasificador ha sido capaz de reconocer el rostro correcto en las tres primeras posiciones, aún cuando la segunda y la tercera cara presentaban ligeras rotaciones con respecto a la original.

Experimento 2. Clasificación de imagen inmersa en ruido

Si ahora la imagen reconstruida estuvo inmersa en ruido, la clasificación será más difícil de llevar a cabo ya que los efectos de la perturbación pueden llevar al clasificador de mínima distancia a cometer errores. Los resultados obtenidos al emplear una imagen ruidosa ($\sigma = 0.05$) se presentan en la siguiente figura:



Figura 6.30 Estimación de los cinco rostros más parecidos (ordenados del primero al quinto en función de la distancia) a la imagen reconstruida (a la izquierda) atendiendo al criterio de mínima distancia. Las imágenes de la base aparecen ordenadas de la primera (la más parecida) a la quinta, y junto a ellas se representa un símbolo (✓) si la identificación fue correcta o (X) en caso contrario.

Como podía ser previsible, en el caso de la reconstrucción de una imagen ruidosa el número de aciertos en la clasificación se ve reducido.

6.4 Conclusiones

En este apartado se ha estudiado el empleo de la Factorización No Negativa de Matrices en la reconstrucción de imágenes de rostros inmersas en ruido aditivo gaussiano, para lo cual se estudió una técnica basada en la Reducción de Código Disperso que permite eliminar las perturbaciones.

Dado que la carga computacional resulta bastante elevada se decidió tratar cada imagen a partir de una serie de regiones de menor tamaño o *patches*. Se plantearon dos posibles vías de reconstrucción: por patches o píxel a píxel, siendo esta última la elegida por arrojar mejores resultados.

Por último, tras realizar diferentes experimentos de reconstrucción y eliminación del ruido, se establecieron comparativas con PCA e ICA para observar las principales diferencias con NMF, que se pusieron de manifiesto al observar el patrón que seguían las bases de cada una de las técnicas de representación de datos.