

Representação Tensorial para Imagens de Faces, Redução de Dimensionalidade e Reconstrução

Tiene A. Filisbino^{1,2}, Victor Vescovini², Gilson A. Giraldi¹, Carlos Eduardo Thomaz³

¹Laboratório Nacional de Computação Científica
CEP 25651-075 – Petrópolis – RJ – Brasil

²Instituto Superior de Tecnologia em Ciência da Computação de Petrópolis
CEP 25651-070 – Petrópolis – RJ – Brasil

³Centro Universitário da FEI, Departamento de Engenharia Elétrica
CEP 09850-901 – São Bernardo do Campo – SP – Brasil

paratiene@yahoo.com.br, vitor.vesconvine@hotmail.com

gilson@lncc.br, cet@fei.edu.br

Abstract. *The main goal of this paper is to discuss some aspects of the Concurrent Subspaces Analysis (CSA) method, used for dimensionality reduction in tensor spaces, when applied to the FEI image database. A total of 14 colorful images for each of 200 individuals compose this database. So, we firstly review some background for tensor algebra followed by the CSA technique presentation. Next, we apply the CSA to the FEI database with special focus on convergence, mean square error and reconstruction. The obtained results are visually suitable and the convergence rate is similar to the one found in the CSA literature.*

Resumo. *O objetivo principal deste trabalho é analisar o comportamento da técnica de redução de dimensionalidade para espaços tensoriais, denominada Concurrent Subspaces Analysis (CSA), para análise da base de imagens de faces do Centro Universitário da FEI. Esta base é constituída por 200 indivíduos, com 14 fotografias coloridas para cada indivíduo. Assim, primeiramente, revisamos alguns elementos da álgebra de tensores necessários para a apresentação do CSA. Em seguida, aplicamos o CSA para a base da FEI com foco na convergência, erro médio quadrático e reconstrução. Os resultados obtidos são visualmente satisfatórios, e a taxa de convergência é semelhante àquela obtida em outros trabalhos da literatura.*

1. Introdução

Redução de dimensionalidade é uma etapa fundamental para os métodos de classificação e mineração de bancos de dados de imagens devido ao tempo de processamento, particularmente em aplicações envolvendo métodos de aprendizagem estatística. Neste caso, o fluxo de dados segue as seguintes etapas: (a) Redução de dimensionalidade; (b) Escolha de um método de aprendizagem para computar um classificador; (c) Reconstrução; ou seja, visualização no espaço de imagens original dos dados representados no espaço de dimensão reduzida. Neste trabalho vamos nos concentrar nas etapas (a) e (c).

No caso da redução de dimensionalidade para bases de dados que podem ser eficientemente representados por pontos em \mathfrak{R}^n , o método tradicional para redução de dimensionalidade é a Análise de Componentes Principais (*Principal Component Analysis-PCA*) [Fukunaga 1990]. Este é o típico caso de imagens em tons-de-cinza, onde para cada imagem corresponde uma matriz $I \in \mathfrak{R}^{m_1 \times m_2}$; ou seja, uma matriz com m_1 linhas e m_2 colunas, na qual cada elemento $I(m, n)$ é a intensidade do pixel correspondente. Desta forma cada dado; mais especificamente, cada imagem $I_i \in \mathfrak{R}^{m_1 \times m_2}$, $i = 1, 2, \dots, N$, do banco de imagens, pode ser representada como um vetor coluna único $\mathbf{x}_i \in \mathfrak{R}^{m_1 \cdot m_2}$ (onde utilizamos ponto "." para representar multiplicação usual de números inteiros).

Porém, podemos ter bancos de dados contendo várias fotos coloridas para cada individuo cadastrado; portanto, com 3 componentes para cada pixel. Neste caso, a representação mais natural para os dados é obtida utilizando-se matrizes generalizadas, denominadas tensores em [M. Alex O. Vasilescu 2002, Xu et al. 2008]. Seguindo a nomenclatura destas referências, um vetor $\mathbf{x}_i \in \mathfrak{R}^{m_1 \cdot m_2}$ é um tensor de primeira ordem, enquanto que uma matriz usual $A \in \mathfrak{R}^{m_1 \times m_2}$ é um tensor de segunda ordem.

Analogamente, podemos definir matrizes de ordem qualquer, denominadas tensores de ordem $n > 2$, digamos $\mathbf{X} \in \mathfrak{R}^{m_1 \times m_2 \times \dots \times m_n}$ com seus elementos envolvendo agora mais de dois índices: $\mathbf{X}_{i_1, i_2, \dots, i_n}$. Neste caso, o *PCA* tradicional pode ser inadequado para tratar o problema de redução de dimensionalidade por envolver processos pouco eficientes do ponto de vista de esforço computacional e armazenamento em memória. Em particular, a matriz de covariância associada ao *PCA* teria dimensão $C \in \mathfrak{R}^{(\prod_{i=1}^n m_i) \times (\prod_{i=1}^n m_i)}$, o que pode ser restritivo para computadores convencionais [Fukunaga 1990]. Assim, é conveniente usar técnicas de redução de dimensionalidade que trabalham na representação natural das amostras (vide seção 5).

Neste sentido, vários trabalhos foram propostos na literatura envolvendo técnicas de redução de dimensionalidade e reconstrução baseadas em representação matricial [Yan et al. 2005, Ye 2004, Yang et al. 2004] e tensorial para imagens [Shashua and Levin 2001]. No primeiro grupo, encontramos técnicas baseadas em decomposição de valores singulares [Ye 2004], generalização do *PCA* para bases de dados representados por matrizes, método denominado *2DPCA* [Yang et al. 2004, Yan et al. 2005], critérios para encontrar dois subespaços ótimos para projetar cada matriz (imagem) da base; um contido no espaço das linhas e outro no espaço das colunas [Shashua and Levin 2001]. Por outro lado, em [Zhang et al. 2008, Ye 2004] é proposto o método *Coupled Subspace Analysis* que vincula a busca destes dois subespaços via uma função objetivo [Shashua and Levin 2001], onde as incógnitas são matrizes de projeção em lugar de vetores de projeção.

A representação tensorial para imagens foi proposta em [M. Alex O. Vasilescu 2002] utilizando uma técnica baseada em decomposição de valores singulares. Outras técnicas de redução de dimensionalidade nesta linha envolvem o método denominado *Concurrent Subspace Analysis (CSA)* [Xu et al. 2008] e extensões multilineares para a técnica de Análise de Componentes Independentes (Multilinear Independent Components Analysis - MICA) [Vasilescu and Terzopoulos 2005]. Nas aplicações para representação tensorial, encontramos reconhecimento de faces em bases de imagens obtidas com condições variadas de iluminação [Kim and Choi 2007], análise e síntese de movimentos faciais e modelos multilineares para bases de faces obtidas via

laser scans [Vlasic et al. 2005].

Neste trabalho, nosso foco será redução de dimensionalidade para bases de imagens representadas via tensores. O método utilizado será o *CSA*, na forma descrita na referência [Xu et al. 2008], a qual apresenta resultados superiores do *CSA*, em relação ao *PCA*, para a base de faces ORL [Xu et al. 2008].

O objetivo deste trabalho é estudar o comportamento do *CSA* para a base de imagens do Centro Universitário da FEI (<http://fei.edu.br/~cet/facedatabase.html>), a qual é composta por imagens coloridas, frontais e não-frontais, com diferentes condições de iluminação. Esta base foi desenvolvida no contexto do projeto descrito em [Junior and Thomaz 2006]. O presente trabalho terá como foco estudar os seguintes aspectos do *CSA* para a base da FEI: (a) Convergência do *CSA*; (b) Erro médio quadrático em função da taxa de compressão; (c) Reconstrução. Uma vez que pretendemos comparar os resultados com aqueles obtidos em [Xu et al. 2008] para a base ORL, faremos um primeiro experimento utilizando as imagens da base da FEI convertidas para tons-de-cinza e normalizadas para a resolução 56×46 pixels. Neste caso, a base é modelada utilizando-se um tensor $X_i \in \mathfrak{R}^{56 \times 46 \times 14}$, para cada indivíduo i da base: cada indivíduo possui um conjunto de 14 imagens em tons-de-cinza, com resolução 56×46 . Assim, o *CSA* gera matrizes ortogonais $U_1 \in \mathfrak{R}^{56 \times m'_1}$, $U_2 \in \mathfrak{R}^{46 \times m'_2}$ e $U_3 \in \mathfrak{R}^{14 \times m'_3}$, com $m'_1 \leq m_1$, $m'_2 \leq m_2$ e $m'_3 \leq m_3$, para projetar cada tensor X_i obtendo uma representação mais compacta $Y_i \in \mathfrak{R}^{m'_1 \times m'_2 \times m'_3}$. Ressaltamos que o *CSA*, assim como a literatura em geral da área de representação tensorial para imagens, pressupõe que todos os tensores da base possuem a mesma dimensão [Lu et al. 2011].

Um segundo teste é feito utilizando as imagens da base da FEI convertidas para tons-de-cinza, mas na sua resolução espacial original de 640×480 . Neste caso, a base é modelada utilizando-se um tensor $X_i \in \mathfrak{R}^{640 \times 480 \times 14}$, para cada indivíduo i da base, sendo que exploramos apenas a redução de dimensionalidade nos espaços das linhas e colunas para evitar os artefatos verificados quando tomamos $m'_3 < 14$. Nos testes usando imagens coloridas, a base de imagens é modelada utilizando-se um tensor $X_i \in \mathfrak{R}^{640 \times 480 \times 3 \times 14}$, para cada indivíduo i da base, uma vez que cada indivíduo possui um conjunto de 14 imagens, em 3 canais de cor (R, G, B), com resolução 640×480 . Neste caso, também, não vamos considerar redução no número de imagens. Assim, o *CSA* retorna três matrizes de projeção referentes a redução na resolução espacial e na dimensão do espaço de cores. As reconstruções obtidas estão visualmente satisfatórias, apesar do *EMQ* ser, em geral, mais elevado para o banco da FEI em relação às imagens utilizadas em [Xu et al. 2008].

O texto do trabalho está organizado da seguinte forma. A base de imagens utilizada está descrita na seção 2. Para facilitar a compreensão do *CSA*, na seção 3 é apresentada uma introdução à álgebra tensorial destacando os conceitos relevantes. A redução de dimensionalidade e reconstrução, no contexto do *CSA*, são discutidas na seção 4. Em seguida, na seção 5, são analisados alguns aspectos sobre o critério de convergência e complexidade computacional do *CSA*, bem como sua comparação com o *PCA*. Os resultados experimentais são apresentados na seção 6. Finalmente, na seção 7 são apresentadas as conclusões e trabalhos futuros.

2. Base de Imagens

O banco de imagens utilizado neste trabalho foi desenvolvido pelo Centro Universitário da FEI, no contexto do projeto descrito em [Junior and Thomaz 2006], o qual tem como objetivo desenvolver um banco nacional de imagens de faces coloridas para desenvolvimento de pesquisas na área de reconhecimento de padrões em imagens. Embora existam diversos bancos de faces internacionais desenvolvidos por universidades estrangeiras, tais como FERET [Philips et al. 1998] e ORL Face Database [Cambridge 2002], o desenvolvimento de um banco de faces nacional é importante para pesquisas que necessitam da caracterização de padrões faciais inerentes a população Brasileira, a exemplo do projeto envolvendo reconhecimento de pessoas desaparecidas coordenado pela FEI [Thomaz 2007].

A base de faces da FEI foi gerada a partir de 200 voluntários, sendo capturadas 14 imagens coloridas de cada indivíduo totalizando 2800 imagens. Com o intuito de permitir estudos que comparam alterações de imagens mais ou menos sutis, foi utilizado o mesmo número de homens e de mulheres para participar desse estudo; no caso, 100 homens e 100 mulheres. A máquina fotográfica utilizada para a captura das imagens é uma Sony Cybershot digital *DSC-W1* (5.1 Megapixels) que fora adquirida pela FEI especialmente para este projeto. O tamanho das imagens capturadas é de 640×480 pixels no formato JPEG.

Todas as fotos foram tiradas sem flash e o zoom utilizado é de 2.4 vezes, sendo permitidas variações de perfil, expressão facial, e iluminação. A Figura 1 ilustra o esquema de captura de imagens utilizado.

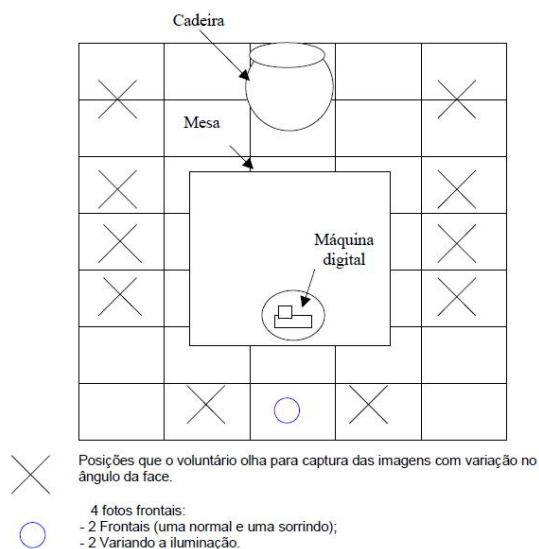


Figura 1. Esquema de aquisição das imagens da base de dados da FEI.

Como pode-se observar por este esquema, as 14 imagens são capturadas em um mesmo ambiente, onde o voluntário fica sentado na cadeira e movimenta a cabeça na direção dos pontos esquematizados na Figura 1, realizando assim a variação desejada no perfil da face. Durante a aquisição das 4 tomadas frontais (símbolo "O" na Figura 1), o voluntário altera também sua expressão facial e o sistema de aquisição altera as condições de iluminação, obtendo resultados como aquele mostrado nas imagens da Figura 2.



Figura 2. Exemplo da base de dados da FEI.

Todas as fotos foram tomadas com fundo branco, com variações de escala na faixa de 10% em relação à imagem média. Foram fotografados indivíduos com idades entre 19 e 40 anos, com aparências distintas, e com a possibilidade de adornos (óculos).

Assim, temos uma base de imagens coloridas, com fundo controlado, pouca variação de escala, mas, muita informação referente a aparência, o que é um diferencial da base da FEI, se comparada a bases tradicionais como FERET [Philips et al. 1998] e ORL Face Database [Cambridge 2002].

3. Álgebra de Tensores

O conceito de tensor tem aplicações em áreas da física (relatividade geral, física de matérias, dentre outras), matemática (geometria diferencial, variedades Riemannianas, etc.) e engenharia (mecânica dos sólidos e fluídos) [B. Dubrovin 1990]. Basicamente, existem duas formas para a definição do conceito de tensor: uma seguindo uma formulação algébrica, baseada em noções de produto tensorial entre espaços e formas multilineares; e, uma outra, esta mais geométrica, baseada em comportamentos de entidades matemáticas sob mudanças de coordenadas [B. Dubrovin 1990]. A escolha destas formulações vai depender da área de aplicação.

No caso de interesse neste trabalho (representação de banco de dados de imagens) o método algébrico é mais simples e direto. Sejam então os espaços vetoriais $\mathfrak{R}^{m_1}, \mathfrak{R}^{m_2}, \dots, \mathfrak{R}^{m_n}$. O produto tensorial destes espaços, denotado por:

$$V = \mathfrak{R}^{m_1} \otimes \mathfrak{R}^{m_2} \otimes \dots \otimes \mathfrak{R}^{m_n}. \quad (1)$$

é outro espaço vetorial, cuja base B natural é dada pelo produto tensorial das bases $\{\mathbf{e}_k^{i_k}, i_k = 1, 2, \dots, m_k\} \subset \mathfrak{R}^{m_k}$ dos espaços em questão, ou seja:

$$B = \{\mathbf{e}_1^{i_1} \otimes \mathbf{e}_2^{i_2} \otimes \dots \otimes \mathbf{e}_n^{i_n}, \quad \mathbf{e}_k^{i_k} \in \mathfrak{R}^{m_k}\}. \quad (2)$$

Neste contexto, definimos um tensor Φ de ordem n como sendo um elemento $\Phi \in V$, ou seja, uma entidade abstrata que pode ser representada como:

$$\Phi = \sum_{i_1, i_2, \dots, i_n} \phi_{i_1, i_2, \dots, i_n} \mathbf{e}_1^{i_1} \otimes \mathbf{e}_2^{i_2} \otimes \dots \otimes \mathbf{e}_n^{i_n}. \quad (3)$$

Por exemplo, no caso de um tensor de segunda ordem temos $\Phi \in \mathfrak{R}^{m_1} \otimes \mathfrak{R}^{m_2}$, definido por:

$$\Phi = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \phi_{i,j} \mathbf{e}_1^i \otimes \mathbf{e}_2^j. \quad (4)$$

Ou seja, dada uma base B , um tensor de segunda ordem fica individualizado pela matriz $[\phi_{i,j}]$. Idem para o caso geral, mas agora temos uma matriz generalizada, com n índices: $[\phi_{i_1, i_2, \dots, i_n}]$. Assim, se fixarmos a base B ; base canônica, por exemplo, podemos representar os tensores por matrizes generalizadas abstraindo a noção algébrica de produto tensorial entre espaços, bem como a interpretação algébrica da expressão (3) como uma forma multilinear [B. Dubrovin 1990]. Assim, um número escalar é um tensor de ordem zero; não necessita índice. Já um vetor é um tensor de ordem um, uma matriz é um tensor de ordem dois (como visto acima), e assim por diante.

Outro conceito importante é o produto tensorial entre operadores lineares em espaços Euclidianos. Sejam então $T_1 : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ and $T_2 : \mathfrak{R}^m \rightarrow \mathfrak{R}^m$, operadores lineares cujas representações matriciais são dadas pelas matrizes:

$$[T_1] = \{b_{i,j}\}, \quad \text{and} \quad [T_2] = \{c_{i,j}\}. \quad (5)$$

É importante observar que estamos usando a notação $[T]$ para distinguir um operador T de sua representação matricial. Então, o produto tensorial entre os operadores T_1 and T_2 , denotado por $\Phi = T_1 \otimes T_2$, é definido pelo novo operador:

$$\Phi : \mathfrak{R}^n \otimes \mathfrak{R}^m \rightarrow \mathfrak{R}^n \otimes \mathfrak{R}^m, \quad (6)$$

$$\Phi(\mathbf{u} \otimes \mathbf{v}) : T_1(\mathbf{u}) \otimes T_2(\mathbf{v}). \quad (7)$$

É possível mostrar que Φ é bilinear e que sua representação matricial é obtida pelo produto de Kronecker entre as matrizes $[T_1]$ and $[T_2]$:

$$[\Phi] = [T_1] \otimes [T_2], \quad (8)$$

o qual, por sua vez, é uma matriz de blocos computada por:

$$[T_1] \otimes [T_2] = \{c_{i,j} \cdot [T_2]\} = \begin{bmatrix} c_{0,0} [T_2] & \cdot & \cdot & \cdot & c_{0,N-1} [T_2] \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ c_{N-1,0} [T_2] & & & & c_{N-1,N-1} [T_2] \end{bmatrix}. \quad (9)$$

Pela definição do espaço tensorial V na expressão (1), fica claro que este é isomorfo ao espaço $\mathfrak{R}^{m_1 \cdot m_2 \cdot \dots \cdot m_n}$. Assim, as noções de produto interno e norma em V podem ser emprestadas do espaço $\mathfrak{R}^{m_1 \cdot m_2 \cdot \dots \cdot m_n}$, como segue:

Definição 1: O produto interno de dois tensores $\mathbf{X} \in \mathfrak{R}^{m_1 \times m_2 \times \dots \times m_n}$ e $\mathbf{Y} \in \mathfrak{R}^{m_1 \times m_2 \times \dots \times m_n}$ é definido como:

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i_1=1, \dots, i_n=1}^{m_1, \dots, m_n} \mathbf{X}_{i_1, \dots, i_n} \mathbf{Y}_{i_1, \dots, i_n} \quad (10)$$

Definição 2: Consequentemente, a norma de um tensor é definida a partir do produto interno pela expressão: $\|\mathbf{X}\| = \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle}$, e a distância entre dois tensores \mathbf{X} e \mathbf{Y} é calculada como:

$$D(\mathbf{X}, \mathbf{Y}) = \|\mathbf{X} - \mathbf{Y}\| \quad (11)$$

Por outro lado, a algebra de tensores é definida a partir de operações que permitem gerar novos tensores a partir de um ou mais tensores dados. Em aplicações na física e matemática as operações mais usuais são contração, adição e produto [Bader et al. 2004]. No contexto deste trabalho, necessitamos do produto tensor *modo-k*, que permite gerar um novo tensor a partir de um tensor \mathbf{X} e de uma matriz A , e do *flattening modo-k*, definidas por [Xu et al. 2008]:

Definição 3: O produto tensor *modo-k*, de um tensor $\mathbf{X} \in \mathfrak{R}^{m_1 \times m_2 \times \dots \times m_n}$ por uma matriz $A \in \mathfrak{R}^{m'_k \times m_k}$ é definido por:

$$(\mathbf{X} \times_k A)_{i_1, \dots, i_{k-1}, i, i_{k+1}, \dots, i_n} = \sum_{j=1}^{m_k} \mathbf{X}_{i_1, \dots, i_{k-1}, j, i_{k+1}, \dots, i_n} A_{i, j}, \quad i = 1, 2, \dots, m'_k. \quad (12)$$

Definição 4: O *flattening modo-k*, de um tensor $\mathbf{X} \in \mathfrak{R}^{m_1 \times m_2 \times \dots \times m_n}$ em uma matrix $X^k \in \mathfrak{R}^{m_k \times \prod_{i \neq k} m_i}$, denotado por $X^k \leftarrow_k \mathbf{X}$, é dado pela expressão:

$$X^k_{i_k, j} = \mathbf{X}_{i_1, \dots, i_n}, \quad \text{onde } j = 1 + \sum_{l=1, l \neq k}^n (i_l - 1) \prod_{0=l+1, 0 \neq k}^n m_0 \quad (13)$$

4. Redução de Dimensionalidade e Reconstrução

Seja agora uma base de dados onde cada ponto amostral pode ser representado por um tensor de ordem n , $\mathbf{X}_i \in \mathfrak{R}^{m_1 \times m_2 \times \dots \times m_n}$, $i = 1, 2, \dots, N$. Sejam também n matrizes de projeção U_1, U_2, \dots, U_n , onde $U_k \in \mathfrak{R}^{m_k \times m'_k}$, sendo $m_k > m'_k$, $k = 1, \dots, n$. Nestas condições a representação de um tensor \mathbf{X}_i no espaço $\mathfrak{R}^{m'_1 \times m'_2 \times \dots \times m'_n}$ (representação em dimensão reduzida) é dada pelo tensor \mathbf{Y}_i computado pela equação [Xu et al. 2008]:

$$\mathbf{Y}_i = \mathbf{X}_i \times_1 U_1^T \dots \times_n U_n^T, \quad i = 1, \dots, N. \quad (14)$$

Consequentemente, pode-se mostrar que a reconstrução é dada por um tensor $\mathbf{X}_i^R \in \mathfrak{R}^{m_1 \times m_2 \times \dots \times m_n}$, dado por [Xu et al. 2008]:

$$\mathbf{X}_i^R = \mathbf{X}_i \times_1 U_1 U_1^T \dots \times_n U_n U_n^T \quad (15)$$

A questão fundamental é qual critério de optimalidade utilizar para obter as matrizes U_1, U_2, \dots, U_n ? A resposta é dada em [Xu et al. 2008] pelo seguinte problema de otimização:

$$(U_j|_{j=1}^n) = \arg \min_{U_j|_{j=1}^n} \sum_{i=1}^N \|\mathbf{X}_i \times_1 U_1 U_1^T \dots \times_n U_n U_n^T - \mathbf{X}_i\|^2, \quad (16)$$

cuja solução tem a seguinte propriedade [Xu et al. 2008]:

Teorema 1: Se $(U_1, \dots, U_{k-1}, U_{k+1}, \dots, U_n)$ são fixas, então a matriz U_k ideal é composta pelos m'_k autovetores principais (autovetores com os maiores autovalores) da matriz de covariância $C^k = \sum_{i=1}^N X_i^k X_i^{kT}$, onde X_i^k é a matriz obtida pelo *flattening modo-k* de \mathbf{X}_i^k isto é: $X_i^k \leftarrow_k \mathbf{X}_i^k$ e $\mathbf{X}_i^k = \mathbf{X}_i \times_1 U_1^T \dots \times_{k-1} U_{k-1}^T \times_{k+1} U_{k+1}^T \dots \times_n U_n^T$.

Este resultado permite desenvolver um algoritmo iterativo, denominado *CSA*, para encontrar as matrizes de projeção ótimas. O algoritmo tem como entrada o banco de imagens e as dimensões da nova representação m'_k , $k = 1, \dots, n$. Em seguida, são inicializadas as matrizes de projeção $U_k^0 \in \mathfrak{R}^{m_k \times m'_k}$, $k = 1, \dots, n$, utilizando matrizes ortogonais por coluna (tipicamente, truncando o número de colunas da matriz identidade).

Algorithm 1 Algoritmo CSA

- 1: Inicializa as matrizes de projeção U_k^0
 - 2: **for** $t = 1, \dots$ to T_{max} **do**
 - 3: **for** $k = 1, \dots$ to n **do**
 - 4: Calcular o produto tensor $\mathbf{X}_i^k = \mathbf{X}_i \times_1 U_1^{tT} \dots \times_{k-1} U_{k-1}^{tT} \times_{k+1} U_{k+1}^{t-1T} \dots \times_n U_n^{t-1T}$
 - 5: Fazer o *flattening modo-k* de tensor \mathbf{X}_i^k para a matriz X_i^k : $X_i^k \leftarrow_k \mathbf{X}_i^k$
 - 6: Calcular a matriz de covariância C^k : $C^k = \sum_{i=1}^N X_i^k X_i^{kT}$
 - 7: Calcular os principais m'_k auto-vetores de C^k , $C^k U_k^t = U_k^t \Lambda^k$, que serão as colunas da matriz U_k^t
 - 8: **end for**
 - 9: **if** ($t > 2$ e $Tr[abs(U_k^{tT} U_k^{t-1})]/m'_k > (1 - \epsilon)$, $k = 1, \dots, n$) **then**
 - 10: **break**;
 - 11: **end if**
 - 12: **end for**
 - 13: Retorna as matrizes $U_k = U_k^t$, $k = 1, \dots, n$.
-

Terminada a inicialização de variáveis e matrizes de projeção, são realizados cálculos envolvendo as operações que aparecem no Teorema 1, as quais são computadas dentro de dois laços. O laço externo é controlado pelo parâmetro T_{max} que representa o número máximo de vezes que as matrizes de projeção U_k^t poderão ser atualizadas. Por outro lado, o laço interno, que atualiza as matrizes U_j , $j = 1, 2, \dots, n$, é montado seguindo a filosofia implícita no Teorema 1: tomamos as matrizes $U_1^t, \dots, U_{k-1}^t, U_{k+1}^{t-1}, \dots, U_n^{t-1}$ obtidas até o momento e, resolvemos o problema de otimização (16) tendo como incógnita apenas a matriz U_k^{t-1} . Desta forma, seguindo a notação de [Xu et al. 2008], seja $h(U_1^t, U_2^t, \dots, U_k^{t-1}, \dots, U_n^{t-1}) = \sum_{i=1}^N \|\mathbf{X}_i \times_1 U_1^t U_1^{tT} \dots \times_k U_k^{t-1} U_k^{t-1T} \dots \times_n U_n^{t-1} U_n^{t-1T} - \mathbf{X}_i\|^2$ então, $h(U_1^t, U_2^t, \dots, U_k^{t-1}, \dots, U_n^{t-1}) \geq h(U_1^t, U_2^t, \dots, U_k^t, U_{k+1}^{t-1}, \dots, U_n^{t-1})$. Assim, a cada passo estamos otimizando o valor da função quadrática h o que é uma condição necessária para atingir uma solução, ainda que local, para o problema de otimização. A Figura 3 mostra um esquema que representa estas etapas.

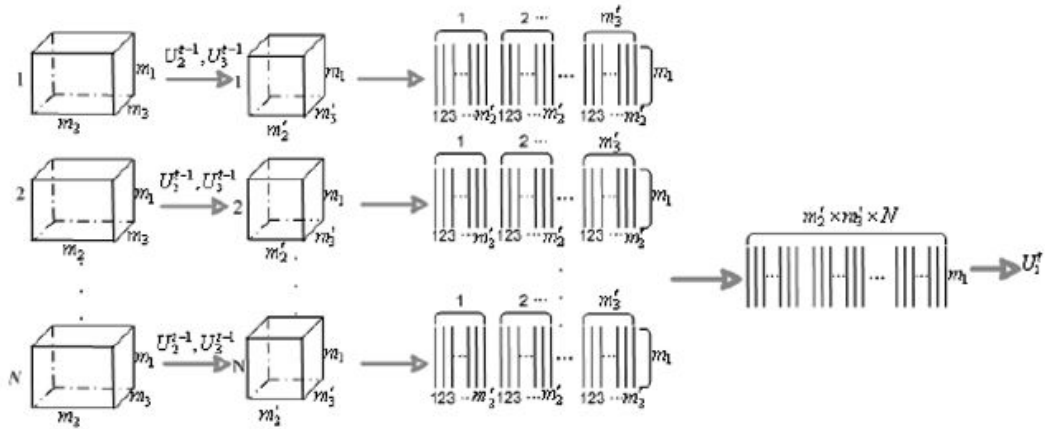


Figura 3. Fluxograma para o CSA considerando dados representados por tensores de ordem 3 (Fonte: [Xu et al. 2008]).

5. Discussão

Nesta seção discutimos alguns aspectos importantes do algoritmo 1. Primeiramente, a expressão:

$$S_k(t) = Tr[abs(U_k^{tT} U_k^{t-1})]/m_k', \quad (17)$$

que aparece na linha 9 do Algoritmo 1, mede a similaridade entre as matrizes de projeção $U_k^t, U_k^{t-1} \in \mathbb{R}^{m_k \times m_k'}$, no seguinte sentido: $0 \leq S_k(t) \leq 1$ e $S_k(t) = 1$ se e somente se $U_k^t = U_k^{t-1}$. Isto é consequência do seguinte resultado.

Lema 1. Uma vez que U_k^t e U_k^{t-1} são ortogonais por coluna, ou seja:

$$U_k^{tT} \cdot U_k^t = I \in \mathbb{R}^{m_k' \times m_k'}; \quad (18)$$

e

$$U_k^{(t-1)T} \cdot U_k^{t-1} = I \in \mathbb{R}^{m_k' \times m_k'}; \quad (19)$$

então, se

$$U_k^{tT} \cdot U_k^{t-1} = I \quad (20)$$

teremos, como consequência,

$$U_k^t = U_k^{t-1} \quad (21)$$

Demonstração: Subtraindo a expressão (18) de (20) vem que:

$$U_k^{tT} (U_k^t - U_k^{t-1}) = O. \quad (22)$$

Procedendo analogamente para (19) e (20) e tomando a transposta do resultado teremos:

$$U_k^{(t-1)T} (U_k^t - U_k^{t-1}) = O. \quad (23)$$

Subtraindo agora (23) de (22) e agrupando termos obteremos:

$$(U_k^t - U_k^{t-1})^T (U_k^t - U_k^{t-1}) = O, \quad (24)$$

o que ocorre se e somente se $U_k^t = U_k^{t-1}$. \square

A condição $S_k(t) \leq 1$ é uma consequência direta do fato de U_k^t e U_k^{t-1} serem ortogonais por coluna. Assim, quando a similaridade é alta; ou seja $S_k(t) \approx 1$, podemos afirmar que o método está próximo da solução e interromper a execução do CSA.

Um outro aspecto importante é a comparação entre o CSA e o PCA tradicional. Do ponto de vista teórico, foi demonstrado em [Xu et al. 2008] que o CSA pode ser visto como uma versão restrita do PCA; ou seja, a matriz de projeção do CSA é dada pelo produto de Kronecker das matrizes de projeção individuais. Formalmente, suponha que os tensores \mathbf{X}_i e \mathbf{Y}_i estão vetorizados dimensão por dimensão gerando \mathbf{x}_i^v e \mathbf{y}_i^v . Visto que $\mathbf{Y}_i = \mathbf{X}_i \times_1 U_1^T \times_2 U_2^T \dots \times_n U_n^T, i = 1 \dots N$, pode-se mostra que $\mathbf{y}_i^v = P^T \mathbf{x}_i^v$ com $P = U_n \otimes U_{n-1} \otimes \dots \otimes U_1 \in \mathbb{R}^{\prod_{k=1}^n m_k \times \prod_{k=1}^n m_k}$ onde \otimes é o produto Kronecker [Xu et al. 2008]. Assim, resolver o problema (16) é equivalente a resolver o problema de otimização:

$$P^* = \arg \min_P \sum_{i=1}^N \|P P^T \mathbf{x}_i^v - \mathbf{x}_i^v\|^2, \quad (25)$$

$$\text{sujeito a :} \quad (26)$$

$$P = U_n \otimes U_{n-1} \otimes \dots \otimes U_1 \quad (27)$$

Para compreender este resultado podemos utilizar a teoria para produto tensorial de operadores apresentada na seção 3. Assim, sejam os operadores de mudança de base T_1 e T_2 :

$$T_1 (\tilde{\mathbf{e}}_1^i) = \mathbf{e}_1^i, i = 1, 2, 3, \dots, m_1, \quad (28)$$

$$T_2 (\tilde{\mathbf{e}}_2^j) = \mathbf{e}_2^j, j = 1, 2, 3, \dots, m_2. \quad (29)$$

Então, substituindo estas expressões na equação (4) obtemos:

$$\Phi = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \phi_{i,j} (T_1 (\tilde{\mathbf{e}}_1^i) \otimes T_2 (\tilde{\mathbf{e}}_2^j)). \quad (30)$$

Usando agora a definição de produto tensorial de operadores, dada pela expressão (7), podemos rearranjar os termos aproveitando a linearidade das operações:

$$\Phi = (T_1 \otimes T_2) \left(\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \phi_{i,j} \tilde{\mathbf{e}}_1^i \otimes \tilde{\mathbf{e}}_2^j \right). \quad (31)$$

Assim, utilizando o fato de que a representação matricial do produto tensorial $T_1 \otimes T_2$ é dada pelo produto de Kronecker entre as matrizes $[T_1]$ e $[T_2]$ (vide equações (8)-(9)), obteremos:

$$[\Phi] = [T_1] \otimes [T_2] \left[\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \phi_{i,j} \tilde{\mathbf{e}}_1^i \otimes \tilde{\mathbf{e}}_2^j \right], \quad (32)$$

ou seja:

$$[\Phi] = [T_1] \otimes [T_2] \begin{pmatrix} \phi_{1,1} \\ \phi_{1,2} \\ \dots \\ \dots \\ \phi_{m_1, m_2-1} \\ \phi_{m_1, m_2} \end{pmatrix}. \quad (33)$$

A expressão (30) corresponde a efetuarmos uma rotação no espaço das linhas e outra rotação independente no espaço das colunas. É óbvio que podemos ter rotações mais gerais no espaço $\mathfrak{R}^{m_1 \cdot m_2}$, o qual é isomorfo ao espaço $\mathfrak{R}^{m_1} \otimes \mathfrak{R}^{m_2}$. O *PCA* procura uma rotação ótima no espaço $\mathfrak{R}^{m_1 \cdot m_2}$ enquanto que o *CSA* procura uma rotação ótima que possa ser expressa como produto de Kronecker de rotações independentes nos espaços das linhas e das colunas, sendo portanto mais restrito em relação ao *PCA*.

Apesar desta desvantagem teórica do *CSA* em relação ao *PCA*, na prática, os trabalhos da literatura têm mostrado resultados superiores para o *CSA* [Xu et al. 2008]. Do ponto de vista de métodos de aprendizagem estatística, uma possível explicação para este fato pode estar no chamado problema de poucas amostras (*small sample size problem*), que aparece quando a dimensão do espaço é muito maior que a quantidade de amostras disponíveis da população [Lu et al. 2003]. As matrizes $[T_1]$ e $[T_2]$ na expressão (33) são obtidas resolvendo dois problemas de autovalores-autovetores distintos, um envolvendo uma matriz de covariância em $C_1 \in \mathfrak{R}^{m_1 \times m_1}$ e outro envolvendo uma matriz de covariância $C_2 \in \mathfrak{R}^{m_2 \times m_2}$. Estas matrizes são montadas utilizando um conjunto de treinamento $\{\Phi_1, \dots, \Phi_K\} \subset \mathfrak{R}^{m_1 \times m_2}$; e, no caso de C_1 utilizando o conjunto $\{l_{11}, l_{12}, \dots, l_{1m_1}; \dots; l_{K1}, l_{K2}, \dots, l_{Km_1}\}$, formada por $K \cdot m_1$ pontos amostrais, onde l_{ij} é a linha j da matriz Φ_i (idem para C_2 , apenas usando as $K \cdot m_2$ colunas). Por outro lado, o *PCA* utilizaria uma matriz de covariância em $\mathfrak{R}^{m_1 \cdot m_2}$, construída utilizando-se o conjunto de pontos amostrais $\{x_1, x_2, \dots, x_K\} \subset \mathfrak{R}^{m_1 \cdot m_2}$, onde x_i é a matriz Φ_i vetorizada como na expressão (33). Fica portanto claro que o *PCA* é mais sujeito ao problema de poucas amostras para espaços de dimensão elevada, como é o típico caso de imagens de faces.

Do ponto de vista de ocupação de memória, observamos que as matrizes de covariância $C^k \in \mathfrak{R}^{m_k \times m_k}$, $k = 1, 2, \dots, n$, enquanto que a matriz de covariância C do *PCA* teria dimensão $C \in \mathfrak{R}^{(\prod_{i=1}^n m_i) \times (\prod_{i=1}^n m_i)}$, o que mostra que o custo de armazenamento em

memória do *PCA* é mais elevado que o do *CSA*. Por outro lado, a solução do problema de autovalores-autovetores para o *PCA* demandaria um custo computacional da ordem de $O[(\prod_{i=1}^n m_i)^3]$ enquanto que a diagonalização das matrizes C^k , $k = 1, 2, \dots, n$ tem um custo computacional da ordem de $O(\sum_{i=1}^n m_i^3)$. Assim, o custo de processamento do *PCA* é, em geral, mais elevado também, como já discutido em [Xu et al. 2008].

6. Resultados Experimentais

Nesta seção vamos apresentar alguns resultados obtidos pela aplicação do algoritmo 1 e da expressão de reconstrução (15), para a base de imagens da FEI, descrita na seção 2. Foi desenvolvida uma implementação em MATLAB, versão R2009A, dos algoritmos citados utilizando as classes para tensores apresentadas em [Bader et al. 2004]. Os experimentos foram realizados em um computador com 12 GB RAM, com sistema operacional Windows 7 Ultimate 64 bits, tendo como processador Intel Core i7 CPU X 980 com 3,33GHz, Placa Mãe uma ASUS P6TD Deluxe e placa de vídeo Nvidia GeForce GTX 480.

Foram feitos dois conjuntos de testes; um para imagens em tons-de-cinza e outro para as imagens no formato *RGB*. O primeiro conjunto de testes é interessante para a comparação com os resultados apresentados em [Xu et al. 2008]. O segundo conjunto permite analisar o algoritmo quando introduzimos mais um grau de liberdade para a redução de dimensionalidade; ou seja, o espaço de cores *RGB*.

Vamos iniciar com os experimentos em tons-de-cinza. O algoritmo tem como entrada a quantidade N de pessoas da base e o número de fotos para cada pessoa. A partir destes parâmetros, usamos as funções do MATLAB para converter as imagens em tons-de-cinza e obter a altura m_1 e largura m_2 de cada foto, que neste caso, são constantes para toda a base. Em seguida, para cada indivíduo i da base, é montado um tensor $\mathbf{X}_i \in \mathfrak{R}^{m_1 \times m_2 \times m_3}$, onde $m_3 = 14$ é o número de fotos para cada pessoa na base da FEI.

Assim, teremos 3 matrizes de projeção U_1 , U_2 e U_3 , as quais são inicializadas como matrizes canônicas $U_k \in \mathfrak{R}^{m_k \times m'_k}$, $k = 1, 2, 3$. A saída do algoritmo 1 é dada pelas matrizes de projeção ótimas, segundo o critério de optimalidade definido pela expressão (16). De posse dos U_k , aplicamos a equação (14) para obter a nova representação em dimensão reduzida ou a equação (15) para obter a reconstrução correspondente. Seguindo os resultados apresentados em [Xu et al. 2008], vamos analisar a similaridade, dada pela expressão (17) e o erro médio quadrático (EMQ) em função da taxa de compressão (TC), definidos pelas equações:

$$EMQ = \sqrt{\sum_{i=1}^N \|X_i^R - X_i\|^2 / N}. \quad (34)$$

$$TC = \frac{Nm_1m_2m_3}{N * m'_1 * m'_2 * m'_3 + (m'_1 * m_1 + m'_2 * m_2 + m'_3 * m_3)}. \quad (35)$$

Se $m'_1 = m'_2 = m'_3 = r$ então a expressão (35) se reduz àquela encontrada em [Xu et al. 2008]. Uma das bases utilizadas em [Xu et al. 2008] foi a base ORL, pre-processada pelo filtro de Gabor com 5 escalas 8 orientações distintas, gerando 40 características para cada imagem. As imagens desta base são originalmente adquiridas em 256

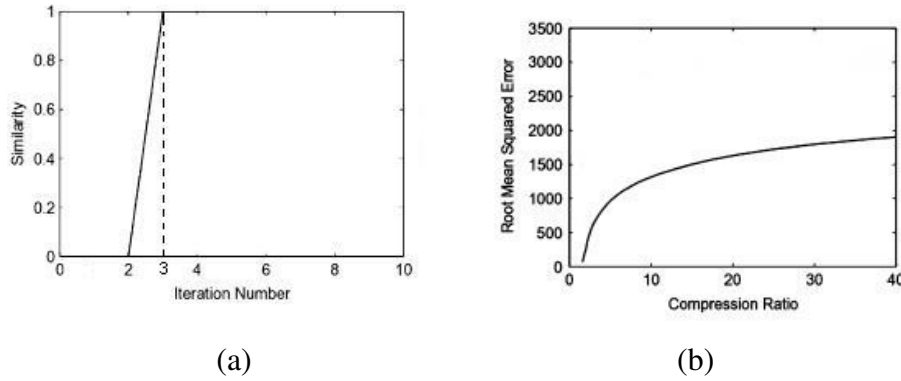


Figura 4. Resultados do CSA para a base ORL, normalizada para 56×46 pixels: (a) Similaridade $S_1(t)$, para $t=1,2,3,\dots$, com $m_1 = m_2 = m_3 = 10$ (Fonte [Xu et al. 2008]). (b) Erro médio quadrático (Fonte [Xu et al. 2008]).

tons-de-cinza, sendo normalizadas para 56×46 pixels nos testes do CSA discutidos em [Xu et al. 2008]. A Figura 4 mostra o comportamento típico destes resultados.

O comportamento da similaridade para a base de imagens da FEI pode ser observado nas Figuras 5, as quais foram geradas para $m_1 = m_2 = m_3 = 10$ com a resolução espacial normalizada para 56×46 pixels, para permitir a comparação direta com os resultados da Figura 4. O eixo vertical mostra a similaridade $S_k(t)$ para $t \geq 1$ (eixo horizontal) e $k = 1, 2, 3$. Observamos que a taxa de convergência do método foi inferior àquela mostrada na Figura 5 apenas para $k = 1$.

A Figura 5 mostra que o algoritmo obtém a solução desejada após quatro iterações ($t = 4$), para $k = 1$ e após três iterações ($t = 3$) para $k = 2, 3$, onde fixamos $\epsilon = 0.001$ na linha 9 do Algoritmo 1, seguindo o que foi feito na referência [Xu et al. 2008]. O fato de termos taxas de convergência distintas para distintos valores de k pode ser justificado observando que:

1. Pela discussão da seção 5, vemos que a expressão (30) corresponde a efetuarmos uma rotação no espaço das linhas e outra rotação independente no espaço das colunas;
2. O fato acima é implementado no CSA tomando a melhor estimativa da matriz U_1 obtida até o momento, dada pela matriz U_1^t , e, resolvendo o problema de otimização (16) tendo como incógnita apenas a matriz U_2 ;

Assim, não há motivo para o algoritmo apresentar taxas de convergência idênticas para distintos valores de k . Por outro lado, o número de iterações necessário para encontrar uma solução para o problema de otimização (16) depende da natureza da base de imagens e da inicialização, podendo assim variar de uma base de dados para outra. Contudo, as taxas de convergência obtidas ficaram bem próximas daquela da Figura 4, o que é um indicativo da corretude da implementação desenvolvida.

A Figura 6 mostra algumas reconstruções obtidas a partir da equação (15). Uma vez que estamos alterando o número de fotos, observamos artefatos decorrentes da projeção no subespaço correspondente.

Os artefatos observados nas imagens da Figura 6 vão interferir no valor do EMQ , representado na Figura 5.(d) o qual foi gerado usando-se $m_1 = m_2 = m_3 = r$ na

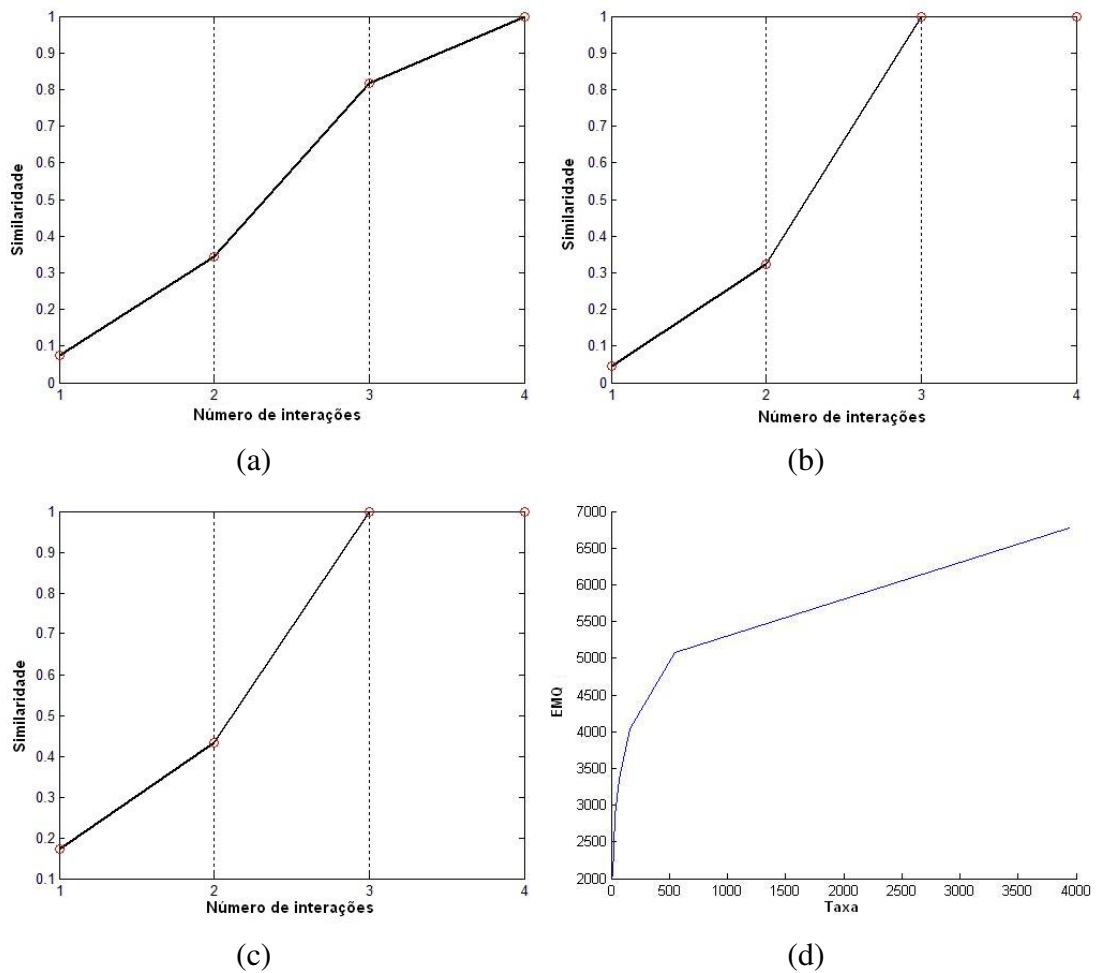


Figura 5. Figura mostrando similaridades $S_i(t)$ para base da FEI usando $mt_1 = mt_2 = mt_3 = 10$ e $\epsilon = 0.001$: (a) Evolução de $S_1(t)$; (b) $S_2(t)$ para $t = 1, 2, 3, 4$. (c) Evolução de $S_3(t)$ para $t = 1, 2, 3, 4$. (d) Erro médio quadrático computado para $r = 2, 4, 6, 8, 10, 12, 14$.

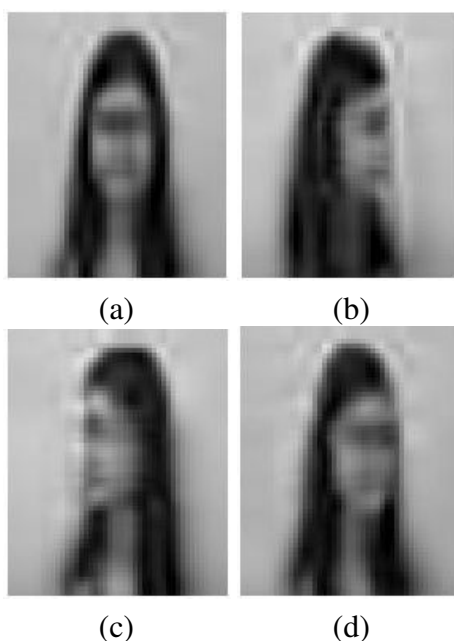


Figura 6. Figura mostrando reconstrução para 4 imagens; uma frontal e tres vistas de perfil, com $m_1 = m_2 = m_3 = 10$.

expressão (35), com $r = 2, 4, \dots, 14$. No caso da base da FEI, além das variações de perfil, temos variações nas condições de iluminação (vide Figura 2). Estes elementos não aparecem na base usada em [Xu et al. 2008] o que é mais uma justificativa para o EMQ mais elevado observado na Figura 5.(d) para a base da FEI se comparado com o gráfico da Figura 4.(b).

Passemos agora para experimentos envolvendo ainda imagens em tons-de-cinza, mas utilizando a dimensão espacial original de 640×480 . A Tabela 1 mostra o número de iterações necessários (t_{max}) para a convergência do CSA nos casos onde $m_1 = m_2 = r$ com $r = 20, 100, 200$, e $m_3 = m_3 = 14$. Assim como no experimento anterior observamos que a taxa de convergência para $S_1(t)$ é ligeiramente maior que para $S_2(t)$, mantendo este padrão de comportamento.

r	t_{max} para $S_1(t)$	t_{max} para $S_2(t)$
20	5	4
100	5	5
200	6	5

Tabela 1. Base da FEI em tons-de-cinza: Redução de dimensionalidade r e número de iterações t_{max} tal que $S_i(t) > (1 - \epsilon)$, onde $\epsilon = 0.001$.

O gráfico da Figura 7 foi gerado a partir dos valores do EMQ , computados pela expressão (34), e as taxas de compressão para $m'_1 = m'_2 = 20, 40, 60, \dots, 480$ e $m'_3 = 14$. Observamos que o comportamento qualitativo dos resultados está coerente; ou seja, a medida que aumentamos r (diminuimos CR) o valor do EMQ diminui.

A Figura 8 mostra o efeito da redução de dimensionalidade na nitidez das imagens. Estas imagens foram geradas, aplicando-se a expressão de reconstrução (15) tendo

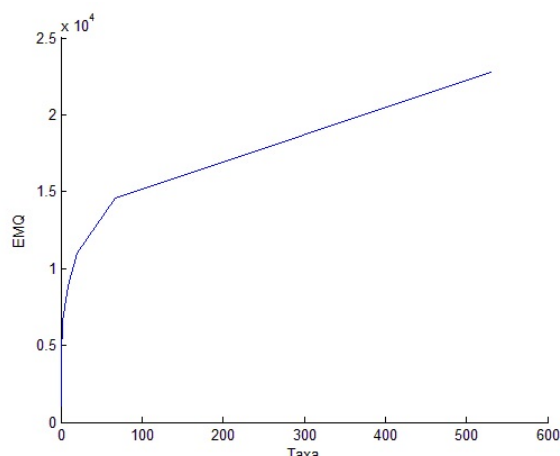


Figura 7. Gráfico $EMQ \times TC$: valores de TC computados para $m'_1 = m'_2 = 20, 40, 60, \dots, 480$ e $m'_3 = 14$.

como argumentos as matrizes U_1 e U_2 geradas pelo Algoritmo 1. O resultado visual é satisfatório, o que é mais um indicativo da corretude da implementação desenvolvida. Outro aspecto relevante é a evolução da qualidade da reconstrução quando aumentamos o valor de r . Por exemplo, comparando a Figuras 8.(c) e 8.(f), onde usamos $r = 60$ e $r = 120$, observamos aumento da qualidade visual. Porém, a perda de qualidade para $r = 60$ é pouco significativa, não interferindo no reconhecimento visual da imagem. Este é um fato importante para aplicações do CSA em reconhecimento de padrões.

A discussão acima pode ser reproduzida para imagens coloridas. Assim, vamos agora considerar a base de dados da FEI, na resolução original de 640×480 , utilizando os três canais de cor RGB ; portanto, $m_1 = 640$, $m_2 = 480$, $m_3 = 3$ e $m_4 = 14$. A Tabela 2 mostra o número de iterações necessário para a convergência do CSA nos casos onde $m_1 = m_2 = r$, com $r = 20, 100, 200$, e $m_3 = 3$, seguido dos resultados análogos para $m_3 = 2$. A quantidade de fotos m_4 não foi reduzida nestes experimentos. Novamente observamos que, de maneira geral, $S_1(t)$ converge mais lentamente que $S_2(t)$.

r	m_3	t_{max} para $S_1(t)$	t_{max} para $S_2(t)$	t_{max} para $S_3(t)$
20	3	5	5	-
100	3	6	5	-
200	3	6	5	-
20	2	4	3	3
100	2	4	3	3
200	2	8	4	3

Tabela 2. Base da FEI em RGB : Redução de dimensionalidade para $m_1 = m_2 = r$ e número de iterações t_{max} tal que $S_i(t) > (1 - \epsilon)$, onde $\epsilon = 0.001$.

O comportamento do erro médio quadrático para estes casos está representado na Figura 9. Observamos que o valor EMQ no primeiro caso, onde não fazemos redução de dimensionalidade no espaço de cor ($m_3 = 3$), é menor que no segundo caso, onde fazemos $m_3 = 2$, o que era esperado.

A Figura 10 mostra algumas reconstruções para imagens coloridas, geradas pela

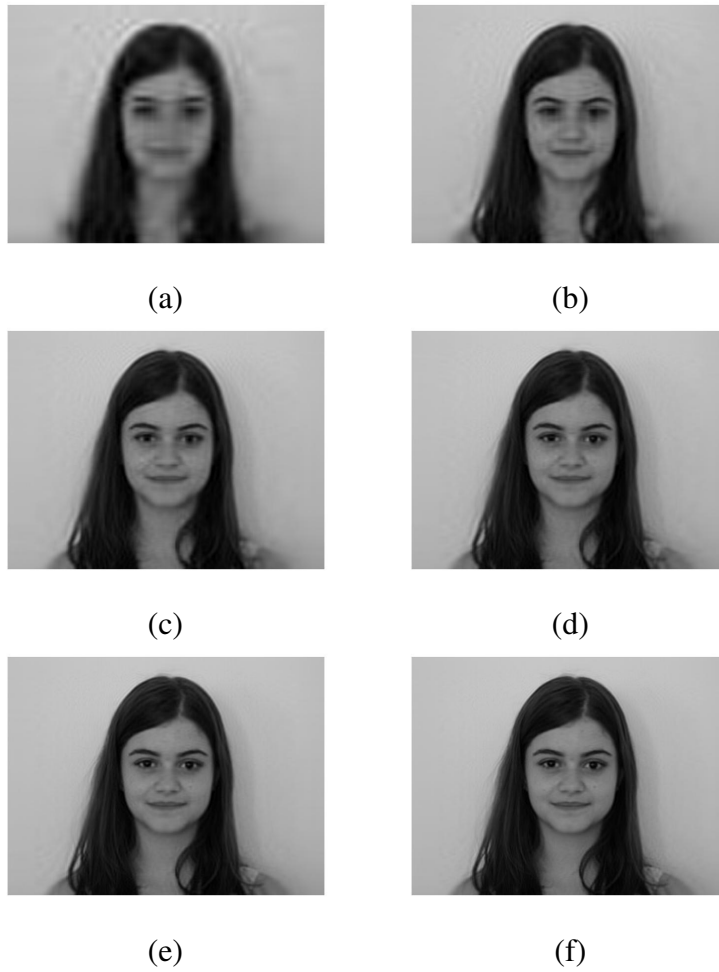


Figura 8. Figura mostrando reconstrução para $r = 20, 40, 60, 80, 100, 120$ para a imagem frontal de um indivíduo da base.

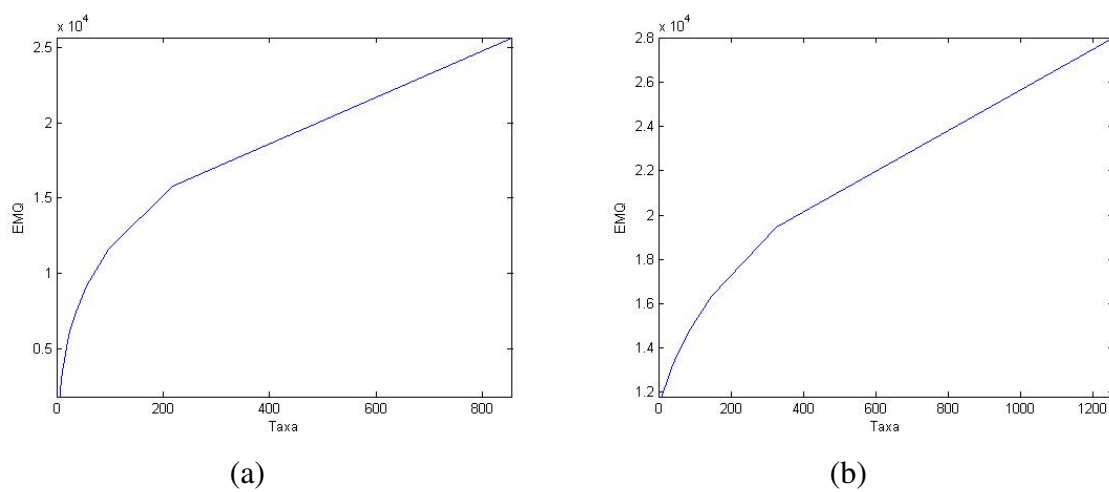


Figura 9. Resultado para base da FEI em RGB: (a) Evolução do EMQ para $mt_3 = 3$. (b) Evolução do EMQ para redução no espaço de cor $mt_3 = 2$.

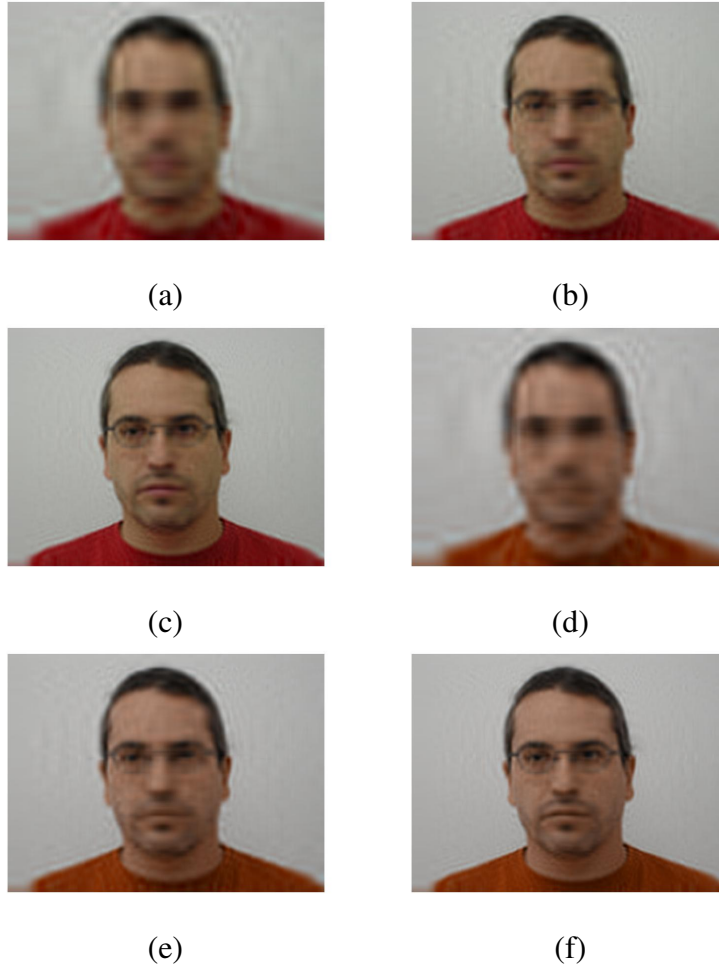


Figura 10. Figura mostrando reconstrução para $m_1 = m_2 = 20, 40, 60$ para a imagem frontal de um indivíduo da base. (a)-(c) Reconstrução para $m_3 = 3$. (d)-(f) Reconstrução para $m_3 = 2$.

expressão (15) tendo como argumento as matrizes U_1 e U_2 e U_3 geradas pelo Algoritmo 1. As Figuras 10.(a)-(c) mostram o efeito da redução de dimensionalidade para $m_3 = 3$ enquanto que as Figuras 10.(d)-(e) mostram este efeito para $m_3 = 2$. Novamente, a qualidade visual dos resultados é bem satisfatória, apesar dos altos valores do EMQ neste caso.

Finalmente, a Tabela 3 fornece a ordem de grandeza dos tempos de execução do laço principal (externo) do algoritmo 1. As duas primeiras linhas desta tabela mostram tempos para a base da FEI, convertida para tons-de-cinza, enquanto que a terceira linha mostra um resultado para as imagens coloridas, onde aplicamos a redução de dimensionalidade no espaço de cor; ou seja, $m_3 = 2$. Observamos que os tempos são diretamente influenciados pelas dimensões da base e pelo valor de r ; o que é coerente, uma vez que estes parâmetros definem as dimensões das matrizes de projeção U_k computadas pelo algoritmo.

Tempo para convergência em segundos do CSA				
Resolução \ r	20	40	100	200
$56 \times 46 \times 14$	1.63	2.27	-	-
$640 \times 480 \times 14$	75.48	-	153.37	297.94
$640 \times 480 \times 2 \times 14$	261.84	-	877.41	9887.70

Tabela 3. Tempo de execução do CSA para base em tons-de-cinza (primeira e segunda linhas) e RGB (terceira linha) para $\epsilon = 0.001$ na linha 9 do Algoritmo 1.

7. Conclusão e Trabalhos Futuros

Redução de dimensionalidade é um tema fundamental para tarefas como reconhecimento e mineração de bancos de imagens coloridas de faces. Nestes casos, é computacionalmente mais conveniente representar a base usando tensores de ordem maior que 2. O Algoritmo CSA, apresentado em [Xu et al. 2008], torna-se uma alternativa promissora para redução de dimensionalidade nestes problemas.

Neste trabalho, analisamos o comportamento do CSA para a base de imagens coloridas de faces mantida pelo Centro Universitário da FEI, com foco na convergência, EMQ e reconstrução. Os resultados obtidos para a convergência foram similares àqueles verificados em [Xu et al. 2008]. As reconstruções obtidas estão visualmente satisfatórias, apesar do EMQ ser, em geral, mais elevado para o banco da FEI em relação às imagens utilizadas em [Xu et al. 2008]. Como trabalhos futuros, ressaltamos a análise de técnicas de reconhecimento e classificação no espaço de características do CSA bem como a generalização de técnicas de aprendizagem de variedades computacionais (*manifold learning*) para dados representados por tensores.

Referências

- B. Dubrovin, A. Fomenko, S. N. (1990). *Modern geometry : Methods and applications*.
- Bader, B. W., Kolda, T. G., and Bader, B. W. (2004). *Matlab tensor classes for fast algorithm prototyping*. Technical report, ACM Trans. Math. Software.
- Cambridge, A. L. (2002). *The orl database of faces*. www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
- Fukunaga, K. (1990). *Introduction to statistical patterns recognition*. 831/836.
- Junior, L. L. O. and Thomaz, C. E. (2006). *Captura e alinhamento de imagens:um banco de faces brasileiro*. *Department of Electrical Engineering, FEI, São Bernardo do Campo, São Paulo, Brazil*, fei.edu.br/~cet/iniciacaocientifica_LeoOliveira_2006.pdf.
- Kim, Y.-D. and Choi, S. (2007). *Color face tensor factorization and slicing for illumination-robust recognition*. In *International Conference on Biometrics*, pages 19–28.
- Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N. (2011). *A survey of multilinear subspace learning for tensor data*. *Pattern Recogn.*, 44(7):1540–1551.
- Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. (2003). *Regularized discriminant analysis for the small sample size problem in face recognition*. *Pattern Recognition Letters*, 24(16):3079–3087.

- M. Alex O. Vasilescu, D. T. (2002). Multilinear analysis of image ensembles: Tensor-faces. 447/460.
- Philips, P. J., Wechsler, H., Huang, J., and Rauss, P. (1998). The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306.
- Shashua, A. and Levin, A. (2001). Linear image coding for regression and classification using the tensor-rank principle. volume 1, pages I–42 – I–49.
- Thomaz, C. E. (2007). Modelagem e reconstrução de imagens de face de crianças e pessoas desaparecidas. <http://fei.edu.br/~cet/projeto-CNPq-Universal2007.pdf>.
- Vasilescu, M. A. O. and Terzopoulos, D. (2005). Multilinear independent components analysis. In *IEEE COMP. COMP. VIS AND PAT. REC. (CVPR)*, pages 547–553.
- Vlasic, D., Brand, M., Pfister, H., and Popović, J. (2005). Face transfer with multilinear models. *ACM Trans. Graph.*, 24:426–433.
- Xu, D., Zheng, L., Lin, S., Zhang, H.-J., and Huang, T. S. (2008). Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis. 1051/8215.
- Yan, S., Xu, D., Zhang, L., Li, M., Ma, W., Liu, Z., and Zhang, H. (2005). Parallel image matrix compression for face recognition. pages 232 – 238.
- Yang, J., Zhang, D., Frangi, A., and Yu Yang, J. (2004). Two-dimensional pca: a new approach to appearance-based face representation and recognition. *Patt. Analysis and Mach. Intell., IEEE Trans. on*, 26(1):131 –137.
- Ye, J. (2004). Generalized low rank approximations of matrices.
- Zhang, L., Xu, D., , Liu, Z., and Zhang, H. (2008). Coupled subspaces analysis. 1051/8215:<ftp://ftp.research.microsoft.com/pub/tr/TR-2004-106.pdf>.