Insert here your thesis' task.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Evaluating performance of an image compression scheme based on non-negative matrix factorization

Bc. Marek Pikna

Department of Theoretical Computer Science
Supervisor: doc. Ing. Ivan Šimeček, Ph.D.

March 7, 2019

Acknowledgements

THANKS (remove entirely in case you do not with to thank anyone)

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on March 7, 2019 . . . .

Citation of this thesis

Pikna, Marek. *Evaluating performance of an image compression scheme based on non-negative matrix factorization.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.

Abstrakt

V několika větách shrňte obsah a přínos této práce v českém jazyce.

Klíčová slova Replace with comma-separated list of keywords in Czech.

Abstract

Summarize the contents and contribution of your work in a few sentences in English language.

Keywords Replace with comma-separated list of keywords in English.

Contents

Introduction **1**

1 Non-negative matrix factorization **5**
- 1.1 Problem definition 5
- 1.2 Problem solution 6
- 1.3 Common NMF applications 8
- 1.4 NMF and compression 12

2 Digital image encoding **15**
- 2.1 Digital images 15

3 Image compression **19**

4 NMF compression scheme **21**

Conclusion **23**

Bibliography **25**

A Acronyms **29**

B Contents of enclosed CD **31**

List of Figures

1.1 Comparison between dimensionality reduction algorithms - *NMF* and *PCA*, as shown in [13]. 10
1.2 Visual display of non-hierarchical properties of NMF. [17] 11
1.3 Visualization of NMF as a compression tool. When using a rank r lower than $n/2$, the total amount of elements in matrices W and H is less than the amount of elements in matrix V. When the rank r reaches the value $n/2$, the amount of elements is the same. . . . 13

Introduction

Data encoding and consequently data compression are both problems which lie at the heart of many modern technologies - digital television, videogames, mobile communications, security cameras and all other kinds of multimedia. As the amount of data only grows in the current world, the quality of compression ends up becoming a very serious problem, since good compression can very significantly reduce the costs of data storage as well as the costs and speed of data transfer.

To put the problem into perspective, in order to store images in the resolutions currently considered as high resolution (1920x1080 pixels, the second most common resolution on desktop devices [3]. When storing an image with these dimensions using an uncompressed standard encoding, the filesize would be almost 6 megabytes. Such a high size impacts many areas, such as speed of transfer or storage costs. Fortunately, modern image compression formats such as *PNG* or *JPG* are able to reduce this size remarkably.

Currently, images contribute to the amount of data on the internet significantly - not only are images a common form of media for professional purposes but some of the currently largest platofrms on the internet are based on image sharing and image hosting. One modern social media platform centered around image sharing had over 67 million new posts each day [21]. Being able to obtain these images fast and in good quality is therefore highly important for both users, just as it is important for the owners of these services to be able to store this data.

In the recent years with growht related to machine learning and similar areas, such as artifical intelligence, new applications of mathematical concepts were discovered. Some of these concepts which are enjoying high success are algorithms related to *dimensionality reduction.* These algorithms aim to reduce the number of random variables under consideration [20].

While these algorithms have been enjoying success mostly in areas such as data mining or machine learning, their nature of reducing the amount of random variables under inspection means that the algorithms are essentially also compression algorithms. One of the algorithms used for dimensionality reduction and the one which this thesis focuses on is *non-negative matrix factorization* (abbreviated as *NMF*). The *NMF* algorithm is currently used in areas such as facial recognition or astronomy. Heart of the algorithm is the factorization of a matrix consisting of non-negative values into matrices.

The research in image compression methods using dimensionality reduction algorithms is currently being performed - another dimensionality reduction algorithm and its potential usage for image compression which has been well researched is the *singular value decomposition* algorithm, for example in [15]. This algorithm, just like the *NMF*, factorizes a matrix - however, without restricting the values to be non-negative. While certain similar research to see whether *NMF* can be used for image compression exists, the works are related to very specific use-case scenarios.

Thus, this thesis aims to analyze the potential of the *NMF* algorithm as a tool for image compression. In order to achieve this, the following points will be explored in the thesis:

- *NMF* and its current applications will be studied (Chapter 1).
- The theory and practice related to digital image encoding and image compression will be explored and described together with modern image compression . (Chapters 2 and 3)

By analyzing these concepts, a proof of concept image compression algorithm using *NMF* will be designed and implemented. By doing so, these issues will be addressed:

- Whether a certain way of representing an uncompressed image is better suited for non-negative matrix factorization.
- Utilizing both subjective as well as objective metrics commonly used for evaluating quality of image compression, the performance of the proof of concept compression scheme will be evaluated.
- How well suited *NMF* is for usage as an algorithm for image compression.

At the end of the thesis, the proof of concept algorithm will be compared to the state of the art image compression algorithms and possible points related to further analysis or will be explored.

The first three chapters are related to the theoretical part of the problem, studying *NMF*, image encoding and image compression. The following chapters are related to the practical part of this thesis - design and implementation of the image compression algorithm and its evaluation.

Chapter 1

Non-negative matrix factorization

This chapter discusses the *non-negative matrix factorization* - defines the problem, describes some of the existing solutions to the problem and offers some observations. By doing so, the basics for the rest of the thesis are provided.

Non-negative matrix factorization as a problem was first formulated by Paatero and Tapper in [18], but the works which have given this problem far more popularity are the works of Lee and Seung [13], where *NMF* was applied to areas of machine learning and artificial intelligence - more specifically to facial recognition and discovering semantic features in encyclopedic articles.

1.1 Problem definition

Let V be a $n \times p$ non-negative matrix, (i.e. with $x_{ij} \geq 0$, denoted $X \geq 0$), and $r > 0$ an integer. Non-negative matrix factorization consists in finding an approximation

$$V \approx WH \tag{1.1}$$

where W, H are $n \times r$ and $r \times p$ non-negative matrices, respectively - meaning all the elements of the matrices are non-negative. In practice, the rank r is often chosen such that $r \ll min(n, p)$. This is due to the reason than in many common applications of non-negative matrix factorization, the information contained in the matrix V is summarized and split into r factors as the columns of W [6].

It should be noted here that the name of the problem might be misleading, as the term *"factorization"* is usually understood more as an exact decomposition, whereas *NMF* is in reality an approximation. Thus, the problem is called *non-negative matrix approximation* in certain other works, such as [23].

1.2 Problem solution

In this section, two of the commonly used algorithms for solving the non-negative matrix factorization problem will be described. The first of these two algorithms will be the algorithm based on *multiplicative updates* as used in [13], where the attention to part-based analysis, simplicity of the *multiplicative updates* and interpretability of the results helped to spread the influence into many other research fields, such as image processing or text processing [9]. Due to these reasons, this will be one of the algorithms described. The second algorithm which will be described is the *alternating least squares* algorithm, which is the earliest algorithm proposed for solving the non-negative matrix factorization problem (positive matrix factorization in the original work) [18].

The reason for choosing these two algorithms is that both of them are very commonly used in practice. Other algorithms exist and are often researched, example being the *projected gradient* method [11]. However, they will not be explored within this thesis.

1.2.1 Multiplicative updates

The algorithm described here is described more thoroughly in [14], a work by Lee and Seung where the algorithms are described in detail together with proving correctness of the algorithms.

In order to find an approximate factorization $V \approx WH$, a way how to quantify the quality of the approximation needs to be defined. Such a metric (or a cost function) can be constructed by measuring the distance between two non-negative matrices A and B. One of the measures provided in [14] is the square of the Euclidean distance between A and B.

$$||A - B||^2 = \sum_{ij}(A_{ij} - B_{ij})^2 \tag{1.2}$$

This distance is lower bounded by zero and vanishes if and only if $A = B$.

By using this cost function, the non-negative matrix factorization can be formulated as an optimization problem:

Problem 1 *Minimize $||V - WH||^2$ with respect to W and H, subject to the constraints $W, H \geq 0$.*

It is shown in [14] that an algorithm cannot realistically solve this problem by finding a global minimum. However, it is possible using various techniques from numerical optimization which make it possible to find local minimum.

Thus, the multiplicative update rules are a compromise between speed and ease of implementation offered in [14] for solving this problem. The multiplicative updates can be described as an algorithm below:

Input: Non-negative matrix V
Output: Non-negative factors W and H

- Initialize W and H as non-negative matrices
- Until $||V - WH||^2$ is minimized, update W and H by computing the following, with n as an index of the iteration:

$$H_{[i,j]}^{n+1} = H_{[i,j]}^{n} \frac{((W^n)^T V)_{[i,j]}}{((W^n)^T W^n H^n)_{[i,j]}} \tag{1.3}$$

and

$$W_{[i,j]}^{n+1} = W_{[i,j]}^{n} \frac{(V(H^{n+1})_{[i,j]}^T}{W^n H^{n+1}(H^{n+1})_{[i,j]}^T} \tag{1.4}$$

Algorithm 1: Multiplicative update algorithm for NMF

It is shown in [14] that the Euclidean distance $||V - WH||$ (and consequently the cost function $||V - WH||^2$) is nonincreasing under these rules. The work by Lee and Seung also considers another possible cost function and proves the convergence of these rules.

1.2.2 Alternating least squares method

Alternating least squares method is the first algorithm proposed for solving the non-negative matrix factorization problem, which was proposed in the work by Paatero. [18] Fixing either of the factors W or H, the problem essentially becomes a least squares problem, which is commonly used in *regression analysis*.

The alternating least squares problem then solves NMF using the algorithm shown in 2.

As the least squares algorithm does not enforce the constraint of non-negativity, all the negative elements in matrices are set to 0 after each evaluation of the least squares problem.

The name of the algorithm reflects its nature where it keeps alternating between solving the least squares problem for one fixed matrix and then the other.

Input: Non-negative matrix V
Output: Non-negative factors W and H

- Initialize W as random dense matrix
- Until a stopping condition:
  (LS) Solve $min_{H \geq 0} ||V - WH||^2$
  (NONNEG) Set all the negative elements of H to 0.
  (LS) Solve $min_{W \geq 0} ||V^T - H^T W^T||^2$
  (NONNEG) Set all the negative elements of W to 0.

Algorithm 2: Basic Alternating least squares algorithm for NMF. [19]

1.3 Common NMF applications

In this section, a short example of a common application of non-negative matrix factorizations will be provided, for the purpose of showing concrete examples so that the reader may become more adjusted to the problem as well as showing certain observations about the properties of non-negative matrix factorization.

1.3.1 Part-based analysis

What has inspired the work of Lee and Seung [13] was the human activity of recognizing objects from basic parts - especially when using human vision which is shown to be designed to detect the presence or absence of features (parts) of physical objects in order to recognize them [5].

Thus, assuming that features of an object would be independent and it would be possible to compile all the features together, an object could be described formally as:

$$Object_i = Part_1(b_{i1}) \text{ with } Part_2(b_{i2}) \text{ with...}, \tag{1.5}$$

where b_{ij} either has the value *present* if part i is present in object j or the value *absent* if part i is absent in object j.

If the possible states *present* and *absent* in the model are replaced by non-negative values, it is possible to signify not only the presence or absence of a feature but also its quantity or significance ($b_{ij} \geq 0$). Thus, mathematically, a description of an object could look like the following:

$$Object_i = b_{i1} \times Part_1 + b_{i2} \times Part_2 + ... \tag{1.6}$$

[9]

When utilizing non-negative matrix factorization, the matrix W can be considered the set of features present in the data and matrix H the set of hidden variables. Non-negative matrix factorization can also be implemented as:

$$v_i \approx W h_i \tag{1.7}$$

Represented this way, the concept of non-negative matrix factorization can be understood intuitively - each column in the original matrix V is a data point. Each column in the matrix W is a basis element and columns of the matrix H give the coordinates of a data point in the basis W. The product matrix WH (the approximation of V) is a linear combination of the column vectors - the features extracted from the data points and its significance in the data point.

This way, features and parts can be extracted from the original data and understood, as shown in the next subsection on the image learning example.

1.3.2 Image learning

Digital image processing is a field which is currently enjoying high popularity when it comes to research. Image processing is the field thanks to which it is possible to extract features from images or recognize various patterns. Principal component analysis *(PCA)* is a dimensionality reduction technique similar to *NMF* commonly used for recognizing faces in images [4] - however, without the non-negativity constraint. It has been shown in [13] that *NMF* can be used in a similar fashion while potentially classifying the data in a way which is easier to be understood.

The Figure 1.1 has been taken from [13], where a database of 2,429 facial images has been taken and used to create a matrix V. By applying both *NMF* as well as *PCA* to the matrix, feature and coeffiecient matrices were created and particular instance of a face was approximately represented by a linear superposition of basis images. The coefficients used in the linear superposition are shown in the montages, where black pixels indicate positive values and red pixels indicate negative values. It can be seen on these montages that while *NMF* can be used for the same purposes as *PCA*, the main strength of the algorithm is that certain parts can be recognized meaningfully - for example parts of a nose and such, whereas in the case of *PCA*, discovering meaning in the matrices is difficult.

It is for these reasons that non-negativity is a powerful and meaningful constraint, as parts are never subtracted from the particular instance (as it can happen in the case of *PCA*), but are only added together [9].

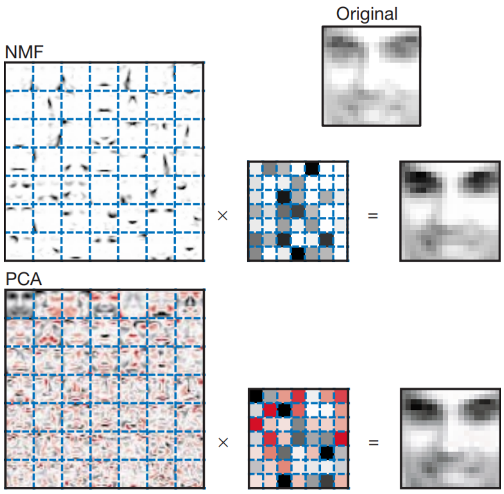


Figure 1.1: Comparison between dimensionality reduction algorithms - *NMF* and *PCA*, as shown in [13].

1.3.3 NMF properties

Previous sections in this chapter have described non-negative matrix factorization, more specifically the definition of the problem, common solutions and an example usage of *NMF*. In this section, certain properties of non-negative matrix factorization will be pointed out. When the proof of concept image compression scheme is designed, the usefulness of these properties for compression will be discussed.

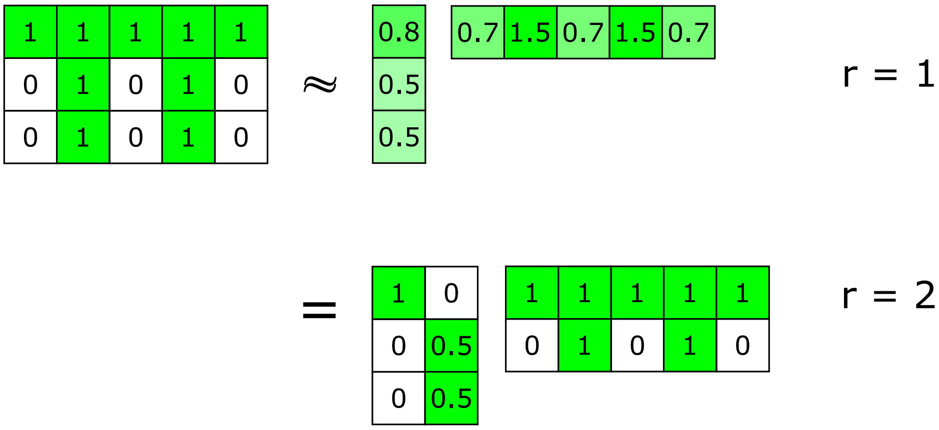
The first property which will be shown is that the solution to non-negative matrix factorization is **not unique**. A matrix and its inverse can transform the two factorization matrices, for example as:

$$V \approx WH = WBB^{-1}H \tag{1.8}$$

If the matrices $\bar{W} = WB$ and $\bar{H} = B^{-1}H$ are non-negative then they form another solution to the *NMF* problem.[24]

Another important property is that the non-negative matrix factorization is not hierarchical, meaning that the factor matrices using rank r can be completely different to those of rank $r + 1$, as shown in 1.2, where choice of rank $r = 1$ provides only approximation of the target matrix V yet rank $r = 2$ makes it possible to calculate $V = WH$ instead of the approximation. Due to these reasons, the results provided by using *NMF* highly depend on choice of the rank parameter.

Figure 1.2: Visual display of non-hierarchical properties of NMF. [17]



However, one of the most important properties to note is that non-negative matrix factorization is very difficult to solve and the existing general algorithms solve *NMF* as an optimization problem. Not only that but it has even been proven that non-negative matrix factorization is an *NP-hard* problem.[22] With certain constraints, it is however possible to solve the *NMF* problem in polynomial time - for example it is shown in [12] that in case the matrix V is symmetric and contains a diagonal principal submatrix of rank r, it is possible to solve the problem in polynomial time $O(rm^2)$.

The last property of non-negative matrix factorization to be noted has been shown in the figures above. The factors extracted are often *sparse* - it is precisely for this reason that interpreting results of non-negative matrix factorization is easy in fields such as image processing or text mining (but many others as well).[7]

1.3.4 Choice of rank r

The choice of rank r is one of the most important problems when utilizing non-negative matrix factorization - as when used for the examples above, choice of rank r changes the amount of features to be extracted in order to approximate the target matrix V.

There are several common methods used for choosing the rank r:

- Trial and error - try different values of r and choose the one performing the best for application at hand.
- Estimate the rank r using various statistical approaches (such as by using *SVD*).

- The use of expert insights.

[7]

1.4 NMF and compression

Most common use-case scenarios of non-negative matrix factorization are related to machine learning, as shown in the previous sections. However, *NMF* can also be looked upon as a lossy compression tool (more on lossy compression in chapter 3), as an original matrix of size $n \times p$ is approximated by the product of two smaller matrices. Assuming the target matrix V is represented in the same way as the factor matrices W and H, if the amount of elements contained in matrices W and H is lower than the amount of elements in matrix V, then *NMF* was used to perform compression.

Whether the amount of elements in factor matrices W and H is lower than the amount of elements in target matrix V depends on the choice of rank r. More specifically, if the dimensions of matrix V are $n \times p$, the dimensions of matrix W $n \times r$ and the dimensions of matrix H $r \times p$, then this requirement can be formalized as:

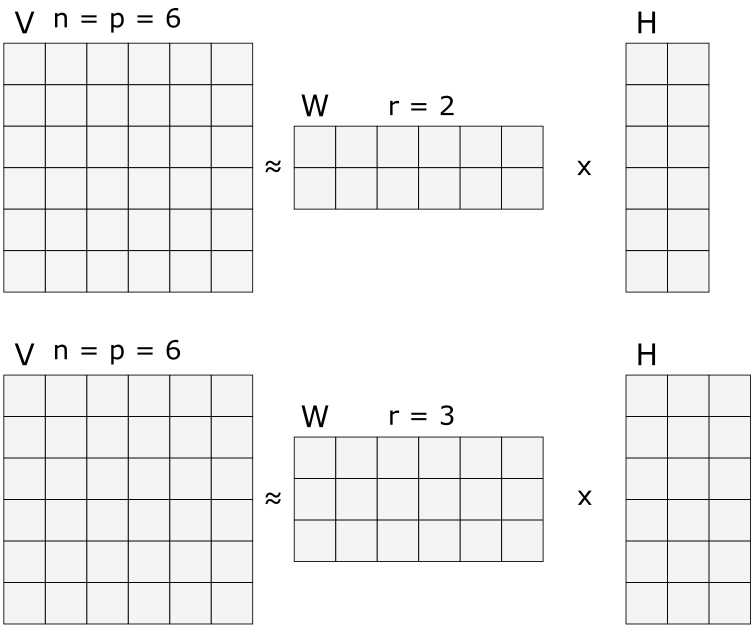
$$\begin{aligned} n \times r + r \times p &< n \times p \\ r(n+p) &< n \times p \\ r &< \frac{n \times p}{n+p} \end{aligned}$$

Considering a square matrix ($n = p$), the following relationship can also be deduced:

$$\begin{aligned} r &< \frac{n \times n}{n+n} \\ r &< \frac{n^2}{2n} \\ r &< \frac{n}{2} \end{aligned}$$

This relationship is also demonstrated in the figure 1.3.

Figure 1.3: Visualization of NMF as a compression tool. When using a rank r lower than $n/2$, the total amount of elements in matrices W and H is less than the amount of elements in matrix V. When the rank r reaches the value $n/2$, the amount of elements is the same.



Chapter 2

Digital image encoding

The second chapter of this thesis is related to digital images and their representations color models. The importance of this chapter is related to creating the image compression scheme and its empirical testing on various different digital image representations. The possible ways of specifying colors in digital images which will be explored are *RGB*, *grayscale* and $Y'C_BC_R$. As this thesis is related to image compression and non-negative matrix factorization and not image processing, only the most important elements of image encoding and color spaces will be explored.

2.1 Digital images

A digital image I is stored as a matrix of *pixels* (abbreviation for a *picture element*). These matrices described as 2D discrete space are derived from analog images in 2D continuous space through the process called *sampling*. More about sampling can be found for example in [25].

The value assigned to a pixel $I[m, n]$ determines its color. The following subsections explore the common color encoding options.

2.1.1 RGB

The *RGB* color model is closely related to the way the human eye perceives colors with the r (red), g (green) and b (blue) receptors in our retinas.[10]. In order to represent a color, components of each color (red, green and blue) are added together. As these components are added together, the *RGB* model is considered to be an additive one.

In order to store image data using the *RGB* color model, the color components need to be quantified. Common way of storing the values of components in a pixel is storing the color intensity value using 8 bits (range [0, 255], where the value 0 indicates no inclusion of the color component and 255 indicates maximum possible inclusion of the component). If all the values of components are equal to 0, the resulting color is black, if all the values of components are equal to the defined maximum value (255 in this case), the resulting color is white. An uncompressed image format which represents images this way is, for example, the Windows BMP.[1]

Thus, encoding an uncompressed image using the *RGB* color scheme with the common 8-bit per component component representation results in each pixel being represented by 24 bits. The size of an image in bytes would then be $width * height * 3$ bytes (not counting the header and other data used by the specific file format).

2.1.2 Grayscale

Grayscale images are images composed exclusively of shades of gray. When using the grayscale model, each pixel therefore carries only the intensity information. Commonly *[CITATION NEEDED]*, grayscale images are stored using 8 bits per pixel. The range of colors represented by these 8 bits goes from black (the value 0) through possible shades of gray to white (255 or another maximum possible value).

It is possible to convert an image stored with the *RGB* model to a a grayscale image by eliminating the hue and saturation information from the image while retaining the luminance. [**?**]

Encoding an uncompressed grayscale image which uses the common 8 bit per pixel representation would therefore create an image of a size of $width * height$, not counting the header and other data used by the specific file format.

2.1.3 $Y'C_BC_R$

Y'CbCr, also written as $Y'C_BC_R$ is a color space commonly used in digital image systems, which is defined by a mathematical coordinate transformation from an associated RGB color space. [**?**] The Y' holds the *luma* value (brightness of an image). The C_B and C_R values are considered the *chroma components*, and represent the color information.

2.1.3.1 Luma

The *luma* value is represented in the $Y'C_BC_R$ model by the symbol Y' and representes the brightness of an image. Y itself is considered to be the *relative luminance*. Relative luminance is a metric of light intensity as it appears to the human eye. The prime symbol (Y') denotes that *gamma correction* has been utilized. Gamma correction is an operation related to nonlinearity of light perception - when twice the number of photons hit a camera sensor, twice the signal is received, denoting a linear relationship. However, the human eye does not perceive change of light in a linear way. Gamma correction thus aims to translate the human eye's light sensitivity and that of a camera. [16] As gamma correction is not an important topic for the rest of this thesis, it will not be explored further.

Luma is calculated as the weighted sum of gamma-compressed $R'G'B'$ components. The prime again represents gamma correction. Luma can be calculated in the following way, as described in [2]:

$$Y' = 0.2126R' + 0.7512G' + 0.0722B' \tag{2.1}$$

2.1.3.2 Chrominance

Chrominance is the signal conveying the color information of a picture, separately from the accompanying luma. the C_B and C_R values represent the blue-difference (and red-difference, respectively) when compared to the neutral, i.e. grayscale. [**?**] Multiple ways of calculating C_B and C_R exist, such as the one for HDTVs in [2]. In digital images, other transformations exist, such as the one used in the JPEG image format:

$$\begin{aligned} C_B &= 128 - (0.168736R') - (0.331264G') + (0.5B') \\ C_R &= 128 + (0.5R') - (0.418688G') + (0.081312B') \end{aligned} \tag{2.2}$$

[8]

2.1.3.3 Use of $Y'C_BC_R$

fgs

Chapter 3

Image compression

fgsfds

Chapter 4

NMF compression scheme

fgsfds

Conclusion

Bibliography

[1] *Microsoft Windows Bitmap File Format Summary.*
URL http://www.fileformat.info/format/bmp/egff.htm

[2] Recommendation ITU-R BT.709-6, 06/2015 ed.

[3] *Desktop Screen Resolution Stats Worldwide.* http://gs.statcounter.com/screen-resolution-stats/desktop/worldwide, 2019. Accessed: 2019-02-20.

[4] Agarwal, M., Agrawal, H., Jain, N., and Kumar, M. *Face Recognition Using Principle Component Analysis, Eigenface and Neural Network.* In 2010 International Conference on Signal Acquisition and Processing, pp. 310–314. 2010. doi:10.1109/ICSAP.2010.51.

[5] Biederman, Irving. *Recognition-by-components: A theory of human image understanding.* Psychological Review, 94:115–147, 1987.

[6] Gaujoux, Renaud. *An introduction to NMF package.* https://cran.r-project.org/web/packages/NMF/vignettes/NMF-vignette.pdf, 2018. Accessed: 2019-02-21.

[7] Gillis, Nicolas. *The Why and How of Nonnegative Matrix Factorization.* arXiv e-prints, arXiv:1401.5226, 2014.

[8] Hamilton, Eric. *JPEG File Interchange Format.* Tech. rep., C-Cube Microsystems, Milpitas, CA, USA, 1992.
URL http://www.w3.org/Graphics/JPEG/jfif3.pdf

[9] Ho, Ngoc-Diep. Nonnegative matrix factorization algorithms and applications. Ph.D. thesis, 2008.

[10] Hunt, R. W. G. The reproduction of colour. Hoboken, NJ: John Wiley, c2004, 6th ed ed. ISBN 04-700-2425-9.

[11] jen Lin, Chih. *Projected gradient methods for non-negative matrix factorization.* Tech. rep., Neural Computation, 2007.

[12] Kalofolias, V. and Gallopoulos, E. *Computing symmetric nonnegative rank factorizations.* Linear Algebra and its Applications, 436(2):421 – 435, 2012. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2011.03.016. Special Issue devoted to the Applied Linear Algebra Conference (Novi Sad 2010).
URL `http://www.sciencedirect.com/science/article/pii/S0024379511002199`

[13] Lee, Daniel D. and Seung, H. Sebastian. *Learning the parts of objects by nonnegative matrix factorization.* Nature, 401:788–791, 1999.

[14] Lee, Daniel D. and Seung, H. Sebastian. *Algorithms for Non-negative Matrix Factorization.* In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, Advances in Neural Information Processing Systems 13, pp. 556–562. MIT Press, 2001.
URL `http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf`

[15] Mathews, Brady. *Image Compression using Singular Value Decomposition (SVD).* `http://www.math.utah.edu/~goller/F15_M2270/BradyMathews_SVDImage.pdf`, 2014. Accessed: 2019-02-21.

[16] McHugh, Sean. *Gamma correction.*
URL `https://www.cambridgeincolour.com/tutorials/gamma-correction.htm`

[17] Miettinen, Pauli. *Lecture notes in Data Mining and Matrices*, 2017.
URL `https://www.mpi-inf.mpg.de/fileadmin/inf/d5/teaching/ss17_dmm/lectures/2017-05-29-intro-to-nmf.pdf`

[18] Paatero, Pentti and Tapper, Unto. *Positive Matrix Factorization: A Non-Negative Factor Model with Optimal Utilization of Error Estimates of Data Values.* Environmetrics, 5:111–126, 1994. doi:10.1002/env.3170050203.

[19] Rezghi, M and Yousefi, Masoud. *A projected alternating least square approach for computation of nonnegative matrix factorization.* Journal of Sciences, Islamic Republic of Iran, 26:273–279, 2015.

[20] Roweis, S. T. and Saul, L. K. *Nonlinear Dimensionality Reduction by Locally Linear Embedding.* In Science, vol. 290, pp. 2323–2326. ISSN 00368075. doi:10.1126/science.290.5500.2323.
URL `http://www.sciencemag.org/cgi/doi/10.1126/science.290.5500.2323`

[21] SCHULTZ, Jeff. *How Much Data is Created on the Internet Each Day?* https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/, 2017. Accessed: 2019-02-18.

[22] SHITOV, Yaroslav. *A short proof that NMF is NP-hard*. 2016.
URL https://www.researchgate.net/publication/303217568_A_short_proof_that_NMF_is_NP-hard

[23] SRA, Suvrit and DHILLON, Inderjit S. *Generalized Nonnegative Matrix Approximations with Bregman Divergences*. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, Advances in Neural Information Processing Systems 18, pp. 283–290. MIT Press, 2006.
URL http://papers.nips.cc/paper/2757-generalized-nonnegative-matrix-approximations-with-bregman-divergences.pdf

[24] XU, Wei, LIU, Xin, and GONG, Yihong. *Document Clustering Based on Non-negative Matrix Factorization*. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR '03, pp. 267–273. New York, NY, USA: ACM, 2003. ISBN 1-58113-646-3. doi:10.1145/860435.860485.
URL http://doi.acm.org/10.1145/860435.860485

[25] YOUNG, Ian T., GERBRANDS, Jan J., VAN VLIET, Lucas J., DATA KONINKLIJKE BIBLIOTHEEK, Cip, HAAG, Den, THEODORE, Young Ian, JACOB, Gerbrands Jan, VLIET, Van, and JOZEF, Lucas. *Fundamentals Of Image Processing*, 1995.

Appendix A

Acronyms

NMF Non-negative matrix factorization

PNG Portable Network Graphics

JPEG

PCA Principal Component Analysis

SVD Singular Value Decomposition

Appendix **B**

Contents of enclosed CD

- `readme.txt` .... the file with CD contents description
- `exe` .... the directory with executables
- `src` .... the directory of source codes
  - `wbdcm` .... implementation sources
  - `thesis` .... the directory of LaTeX source codes of the thesis
- `text` .... the thesis text directory
  - `thesis.pdf` .... the thesis text in PDF format
  - `thesis.ps` .... the thesis text in PS format