Abstract

Digital Image Archive Systems are become well known nowadays. It is widely used in such industrial applications as Medicine, History or Building Industry. Image Archive System is a database which allows to find and retrieve necessary Digital Image and representing it using different devices. In view of Spectral Image research, necessity of Spectral Image Archive System has been appeared.

One of the main problems of this system is large amount of the memory, which is allocated for Spectral Image storage. Therefore, there is a necessity of compression method for reduction of amount of saved information.

One such method is proposed in the current thesis. The method is based on Principal Component Analysis (PCA). PCA is a transformation that reduces dimension of the input vectors. Additionally, the method was improved by means of applying color segmentation (clustering) for the more effective PCA-transformation working. K-means method was introduced for the clustering and then clustering method was adopted for PCA specially. All methods were tested with different parameters. As the result of tests tables and diagrams were obtained. It was proved that PCA is appropriate for spectral image compression. Combination of clustering of colors, manipulating spectral, $L^*a^*b^*$ and a^*b^* color, are presented. Furthermore, principal new clustering method was proposed in this master's thesis for PCA application improvement.

Key Words: Spectral Image, Compression, Archive System, Principal Component Analysis, Clustering Methods, k-means, Compression Ratio, Color Difference ΔE, Root Mean Square Error.

Table of Contents

1 INTRODUCTION

In general, Digital Image is a set of points which are named pixels. Every pixel contains color values and is characterized by horizontal and vertical coordinates (width and height respectively). Color values in every pixel characterize color of the picture in the current pixel.

There are a lot of sources of digital images (e.g. scanners, digital cameras). Images can be saved in the different digital formats on CDs, diskettes or hard drive devices. Digital image sources have a format, resolution and size. Some of pictures are suitable for printing, enlargements or publication on a web page. Others can be suitable for some special representation, which require high quality and right color representations (e.g. photos from another planet or advertisement posters).

Archive Systems ware created for the keeping huge amounts of information (including images). This is a database that allows to store and retrieves digital information which is stored in an information carrier. Different applications can retrieve images from the Digital Image Archive System to represent it on different devices. Such kind of databases are widely applicable in history, medicine, architecture and even social service (persons can has his/her own photo in some security database). Main principles of the Image Archive's work are shown below in Figure 1.1:

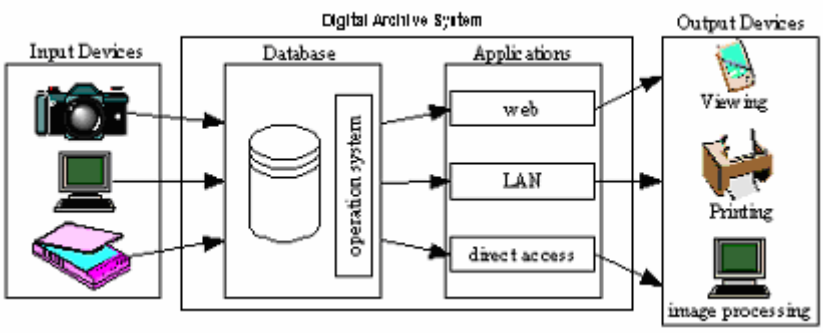

Figure 1.1. Scheme of the Digital Image Archive System

You can see that database obtains source images from different devices. Not only video camera or scanner is used. Other computers can be used as instrument for image creation. Computer graphics, by meaning special graphic tools (e.g. Adobe Photoshop), becomes more widespread now. Database, like any computer device, needs to be supported by some operation systems including system drivers. Different applications (examples are shown in the image) communicate with database for image retrieval. User's part consists of output devices. It is machines, which can be seen and used every day. Database Image Processing System is

the set of tools which are meant for making some transformation (e.g. old icons restoration or pattern recognition for medicine [46])

Images in the Archive System can be saved in different color formats [28]. There are various color space coordinate systems. It can be used to describe colors one of the following ones:

In the RGB model, each color appears in its primary wide band spectral components of red, green and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube shown in Fig.1.2, in which RGB values are at three corners; cyan, magenta, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. RGB model is the hardware-oriented most commonly used in practice for color monitors and a board class of color video cameras [12].

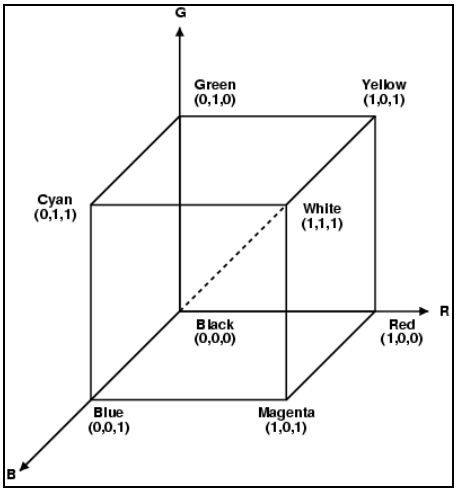

Figure 1.2. Schematic of the RGB color cube.

As shown in the Fig.1.3, cyan, magenta, and yellow are the secondary colors of light or, alternatively, the primary colors of pigments. For example, when a surface coated with cyan pigment is illuminated with white light, no red light is reflected from the surface. That is, cyan subtracts red light from illuminating white light, which itself is composed of equal amounts of red, green and blue light. Most devices that deposit colored pigments on paper, such as color printers and copiers, require CMY data input or perform an RGB to CMY conversion internally.

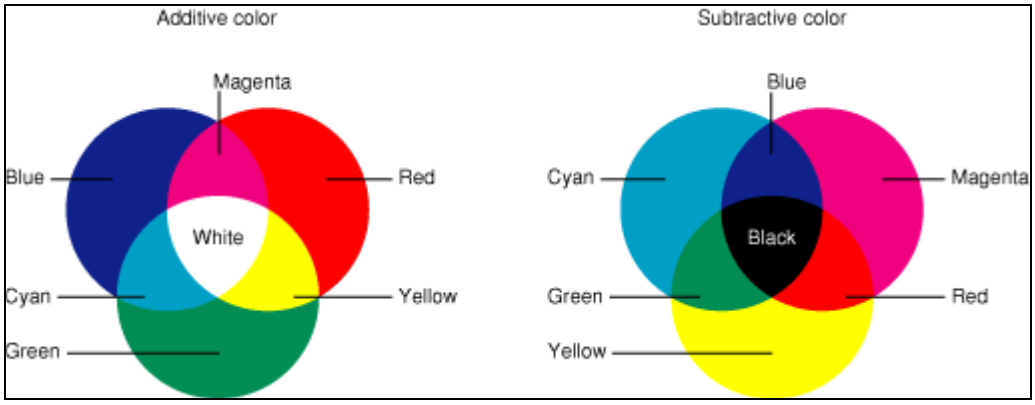

Figure 1.3. Primary and secondary colours of light and pigments.

Tone and color corrections are useful in photo enhancement and color reproduction areas of imaging. The effectiveness of the transformation examined in this section is judged ultimately in print. Since these transformations are developed, refined, and evaluated on monitors, it is necessary to maintain a high degree of color consistency between monitor used and the eventual output devices. In fact, the colors of the monitor should represent accurately any digitally scanned source images, as well as the final printed output. This is the best accomplished with a *device independent color model*. The success of this approach is a function of the quality of the *color profiles* used to map each device to the model and the model itself. The model of choice for many *color management systems* (CMS) is the CIE (Commission Internationale d'Eclairage) $L^*a^*b^*$ model, also called CIELAB (CIE [1987], Roberson [1977])

In a light of problem of right color reproduction there was created one of the best formats for color rendering. It is Spectral Color Format. A Spectral Image is an image where each pixel contains information about the spectral reflectance of the imaged scene. Spectral images carry information about a number of spectral bands: from three components per pixel for color images to several bands for spectral images. Spectral imaging is relevant to several domains of application, such as remote sensing [52], astronomy [45], physics, analysis of museological objects [27], [13], cosmetics, medicine [10], high-accuracy color printing [5], [6], or computer graphics [41]. Spectral image acquisition systems are complex and expensive, limiting their current use mainly to remote sensing applications. Spectral scanners are mostly based on a point-scan scheme [16], [29], [30], and are thus too slow for our applications. This format will be explained more detail at the chapter 2.1 of this thesis.

Necessity of the creation of the Spectral Image Archive System arose as a result of Spectral Image Format used. This is database for storage and retrieval Spectral Images.

One of the main differences between Spectral Image and Digital Image Archive System is the fact that Spectral Images need a lot of memory allocation for the storing. That is why it has to be created compression method adopted for Spectral Images. Method, which allows to reduce amount of allocated memory with minimal information losses. It was considered Principal Component Analysis as the basis. PCA is the transformation, which allows to reduce dimension of the input vectors by using new coordinate system projection. Moreover, for the better method work, PCA-transform was improved by combination of Subspace Clustering method. It is a new method more appropriated for application of PCA.

The main aims of current research were:

Application of Principal Component Analysis for the Spectral Image Data.

Comparison of losses dependency from dimension of the transformation.

Improvement of PCA-transformation for the ΔE reduction by clustering.

Propose another clustering method for the reduction of the ΔE.

Estimation of compression losses and ratio at the every step of the method improvement.

Compare proposed method with conventional methods.

Suggest areas where the proposed method is the most suitable application.

Propose improvements for the methods.

2 SPECTRAL IMAGE ARCHIVE

Current chapter explains to reader meaning and structure of spectral image data and Archive Systems.

2.1 Spectral image data

Color is the sensation of human caused by a stimulus ϕ_λ of electromagnetic radiation within the wavelength range from approximately 380 nm to 780 nm. For a reflection type image, the spectral stimulus ϕ_λ can be calculated by multiplying the spectral power distribution S_λ of the light source with the spectral reflectance $r_\lambda(x, y)$ of the image.

$$\phi_\lambda = S_\lambda \cdot r_\lambda$$

With in the human eye, such a stimulus is received by three types of cones with different spectral sensitivities s_λ, m_λ and l_λ resulting in three values of the standard observer, X, Y and Z. Mathematically, this can be described by the following equations:

$$X = k \cdot \int_{380nm}^{780nm} \phi_\lambda \cdot x_\lambda \, d\lambda ,$$

$$Y = k \cdot \int_{380nm}^{780nm} \phi_\lambda \cdot y_\lambda \, d\lambda ,$$

and

$$Z = k \cdot \int_{380nm}^{780nm} \phi_\lambda \cdot z_\lambda \, d\lambda ,$$

where x_λ, y_λ and z_λ are color-matching functions (vectors) of a CIE standard observer[5] and k - light losses coefficient during way from illuminant to sensor ($k \le 1$).

Two stimuli are considered to appear identically, if their tristimulus values X, Y and Z are equal [9]. This does not imply that the spectral stimuli themselves are identical. A color match of spectrally different stimuli is called a metameric match, and the two colors are called metameric colors.

A TV set for example uses three phosphors to generate metameric matches to the desired natural colors. Therefore, the spectrum generated by the CRT usually is different from the natural spectrum. Similarly, the reflectances of colors produced by conventional printer are

usually different from the reflectances of the original colors. While metameric matches are sufficient for some applications, there are several problems inherent to this approach, if color accuracy is desired [17], [25].

First, a metameric reproduction is illuminant dependent. Therefore, a metameric match is not sufficient if the reproduction is viewed under a variety of illuminants. Imagine the repaired finish of a green car to become a patchwork of green and brown under artificial illumination.

Second, a metameric reproduction is observer-dependent. The reproduced color and the original color only match as long as the standard observer is considered. A human observer, however, usually departs slightly from the standard observer, causing a mismatch between the original and the reproduced color.

Finally, the image capture process of a three channel scanner is document-dependent. This dependency is caused by the deviation of the sensitivities from spectral matching curves. The only to account for these deviations is to characterize the device for every new type of medium (paper, colorants).

Spectral imaging was proposed by several authors to avoid the aforementioned problems of metameric reproduction [17], [25], [18], [8], [13], [7], [23], [24], [15] and [5]. A Spectral image contains the complete spectral characteristics of an image, instead of its tristimulus values. Therefore, it is independent of both the illuminant and the observer.

In practice, the continuous reflectance spectrum is sampled at a large but finite number of wavelengths (e.g. 81 sample values). This is possible because of the smoothness of the sensitivities of the human eye.

A spectral image therefore can be described as a three dimension array $r_\lambda(x, y)$ that can be visualized as a set of separation images. At each pixel location (x_0, y_0), the reflectance vector given by $r_\lambda(x_0, y_0)$ describes the reflectance spectrum of the original image at this location. An image at a given wavelength λ_0 is called a reflectance image $r_{\lambda_0}(x, y)$ [49].

2.2 Spectral Archive Systems

Cataloging and documenting collections are fundamental mandates for all museums. Equally fundamental are educating the public about objects which comprise museum collections and providing access to documentation for these objects. The National Gallery of Art, Washington, DC (NGA) [35], The Museum of Modern Art, New York (MoMA), The Louvre,

Paris [26], The National Gallery, London [36], The State Hermitage Museum, Saint-Petersburg, [51] and The National Museum of Japanese History, Rekihaku [37] they have initiated a research program to develop a scientifically-based, color imaging system which museums, libraries, archives, and similar institutions can use to create digital archives with the highest quality images and most accurate documentation for their collections. More carefully information about this works you can find in R. S. Barnes's webpage [4].

At this time, many digital imaging systems capture film-based or chip-based (digital camera) interpretations of color but not physical measures of color, often rendering inaccurate images and requiring intensive, visually based image editing. These limitations are alleviated by capturing spectral images, that is, images that record the reflected light from a work of art as a function of wavelength. Known as multi-channel visible spectrum imaging, MVSI, these archives will achieve higher information content for scholars and students, significantly reduce image editing, and serve as a valuable tool for conservation science.

Today, as never before, museum, libraries, and archives rely increasingly on digital images to provide access to and preserve their collections. Currently the vast majority of digital image archives are produced in the trichromatic, 24-bit RGB TIFF format. Even if a museum uses the principles of color management during capture and image editing to improve color accuracy, large errors can still occur as a result of the inherent design of camera sensors, whether analog, digital, or chemical.

Consequently, there is an urgent need for color-accurate, digital-image archives that have a scientific basis, meaning a measurement of the physical properties of a work of art. A scientific-based digital archive has significantly higher information content than archives based on images obtained from digitizing conventional photographs or typical direct digital photography. Greater information content, properly applied, can facilitate accurate display using a calibrated monitor on the World Wide Web, color accurate documentation, and opportunities for color-managed printed documents. A scientific-based digital archive requires little or no visual editing, thereby greatly improving the efficiency of workflow during the image capturing process.

Current commercial digital-imaging systems do not have a scientific basis. They are the digital equivalent to conventional film photography, designed to produce pleasing rather than scientific images. Several custom systems have been built to overcome commercial system limitations. They use two different approaches: colorimetric and spectral. The colorimetric

image archive is closely related mathematically to the human visual system. Simply, these images "see" color the same way that humans do. These systems can achieve high color accuracy. The VASARI scanner (of which three have been built for the National Gallery, London [32], [47], [48] [33]; the Doerner Institute, Munich and the Ufizzi Gallery, Florence [49]), the MARC I camera (used by the National Gallery, London and the Doerner Institute, Munich), and the IBM ProSOOO (used by the Vatican) are all examples of equipment which capture colorimetric images. These colorimetric systems can be used to track long-term color changes, provide color-accurate documentation for conservation purposes, and meet other Web-based needs. However, they still lack the ability to measure the fundamental color character of an object.

The second approach to scientific-based digital imaging records spectral information. The light reflecting from or transmitting through an object as a function of wavelength is the only truly fundamental color measurement of a work of art. This area of research is known as spectral imaging and is being used extensively for astronomy and remote sensing where a number of bands throughout the electromagnetic spectrum are imaged. Research concentrating in the visible region[40] has been limited to some universities and research facilities including Color Research Group: University of Joensuu; Ecole Nationale Superieure des Telecommunications, Paris; Aachen University of Technology, Germany; Chiba University, Japan; and the Munsell Color Science Laboratory at RIT. The National Gallery, London and the Centre de Recherche et de Restauration des Musees de France, Paris have expressed great interest in imaging paintings. Although software is currently optimized for colorimetric purposes, it is also possible to estimate spectral data.

The advantages of a spectral image database in comparison to either a trichromatic or near-colorimetric are numerous. Most notably, spectral archives do the following:

Provide a fundamental description of an object;

Define and visualize a work of art’s color under any illuminating and viewing conditions in order to analyze the effect of exhibition lighting on how a painting is perceived and on the quality of painting;

Achieve the highest level of color accuracy;

Allow precise comparison of images to assess color change in an object over time, which allows scholars and conservators to make reasonably accurate simulations of “unfading” (in conjunction with experiments of accelerated aging) and cleaning;

Achieve accurate imaging of fluorescent materials common in modern works of art; and

Enable data from images to be fused with other imaging data such as x-rays, infra-red, and ultraviolet images, to gain greater understanding of the construction of the work of art.

A spectral archive provides all the features of any scientific-based archive. The spectral archive has the additional value of a powerful analytical tool for conservation science. For example, a spectral image of a work of art can be compared with a database of spectrally measured pigments to non-destructively determine what pigments were used in the work. In the future, the spectral imaging technique can be enhanced to include measurement and imaging of spatial properties such as gloss and impasto. Having spectral images will facilitate conservation science and conservation practice, assembly of highly color-accurate image archives, and rendering of highly accurate color reproductions via printing, publishing, and viewing on the Web.

The goal of where this study belongs to is to create an imaging system that stores spectral images, retrieve and represent it on the different devices. During representation process system allows to make restoration process for dye degradation model, simulate different light sources and adopt representation variant for profiles of output devices. Main parts of the Spectral Image Archive System are depicted in Figure 2.1. ImSpector V7-1/2-25 imaging spectrograph was employed in the project of Color Group.

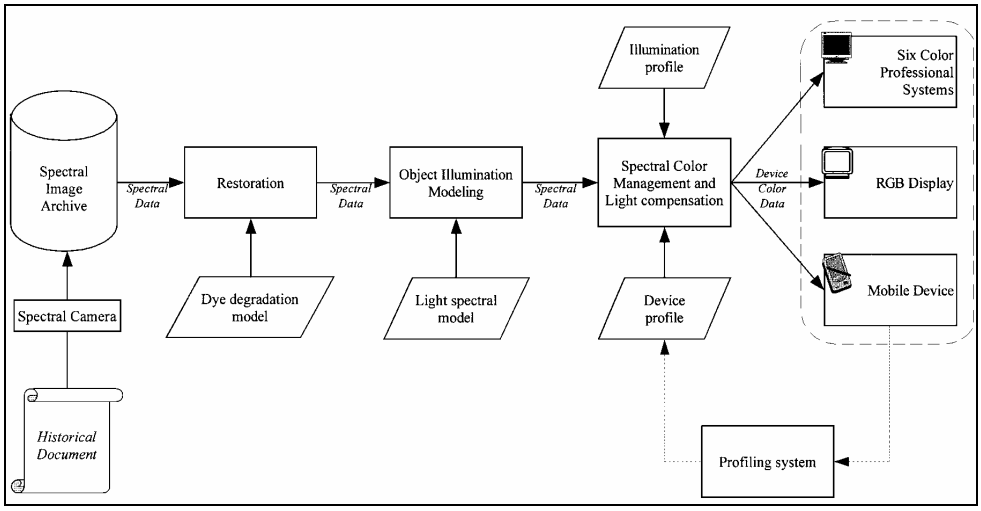

Figure 2.1. Scheme of the Spectral Image Archive System proposed project "Spectral Image Archive System" at the Color Research Group.

3 DESCRIPTION OF THE COMPRESSION METHOD

3.1 Compression methods for spectral images

This chapter introduces compression methods which were proposed for spectral images. The simplest approach would be a compression of the reflectance images $r_{\lambda_i}(x,y)$ for each wavelength λ_i, independent from each other.

Using this approach, only spatial redundancies are removed. This approach is frequently used to encode color images, e.g. in the TIFF standard [53], where each color component is encoded separately.

Depending on the number of sample values used to describe the reflectance spectrum, this basic approach would require the encoding of 81 separation images. Compared to the three channels used for the encoding of a conventional image, this is equivalent to a data enlargement factor of 81/3 = 27, which is unacceptable.

Entered here, data enlargement factor is a coefficient for comparison between encoding of conventional image and spectral one. A multispectral image requires much more data than the same image using conventional technology (e.g. 31 sample values instead of 3 colorimetric values, corresponding to a data enlargement factor of about 10).

The data enlargement factor can be drastically reduced if spectral redundancies are considered. To limit the computational complexity of the compression algorithm it will be consider a two step compression.

First, a spectral compression is applied to each reflectance spectrum $r_\lambda(x_0,y_0)$ to eliminate spectral redundancies. This can be described as a transformation of the spectral image $r_{\lambda_i}(x,y)$ with $1 \le i \le n$ into a compressed image $C(x,y,i)$ with $1 \le i \le m$, that may have a reduced dimensionality $m \le n$. For example, the lossless transformation of the spectral image $r_\lambda(x,y)$ into to the sampling images $S(x,y,i)$ reduces the dimensionality considerably. Likewise, the data enlargement factor is reduced to approximately 5 for a 16-channel spectral scanner. Unfortunately, this transformation is dependent on the spectral scanner and not suitable for a general approach.

The spectral compression is then followed by a spatial compression. Spatial image compression methods have been studied for a long time. The remainder of this paper will therefore deal with spectral compression only.

It is expected, that spectral and conventional three-channel imaging will be used case by case in the future. The main reason is that the color quality achievable with the conventional three-channel approach is sufficient for the majority of imaging applications, like business graphics or non-professional imaging applications. Only applications that require utmost color accuracy, like fine arts archiving, medical diagnostics or professional image reproduction will require the quality of spectral imaging.

Other reasons for the coexistence of spectral and conventional imaging are the lack of spectral output devices and the inability of most of today's image processing software to deal with spectral images.

The combined use of spectral and conventional imaging requires a compatibility of a spectral image format to conventional images. Keusen proposed the spectral high definition Spectral Color description format (SCD) [21].

The term SCD does not describe a single encoding scheme, but rather a class of different encoding methods. These methods apply a basis system transformation to a reflectance vector to decompose the reflectance into its visible and metameric subspace (components $V_1, V_2, V_3, M_4 \ldots M_m$) for one illuminant. In the following, this illuminant will be called the SCD basis illuminant.

The visible subspace is related to the SCD basis illuminant, and can be described by any three channel description. For example the visual components V_1, V_2 and V_3 can be chosen as the standard sRGB values R, G and B or the tristimulus values X, Y and Z.

If the visible components can be easily separated from the remaining components, compatibility is achieved between the SCD and conventional imaging methods. Figure 3.1 demonstrates the idea of SCD methods and gives examples of possible implementations. The first three components are used to encode the visual information for the SCD basis illuminant, while the remaining components describe the metameric information.

The SCD does not only provide compatibility to conventional imaging, but also provides a method for compression if the metameric components are chosen carefully. Keusen's approach was to use principal components to span the metameric subspace. The advantage of this approach is that the coefficients for the higher metameric components tend to have smaller energy. Therefore, the SCD offers an easy way for a lossless encoding, although this approach is not especially effective.

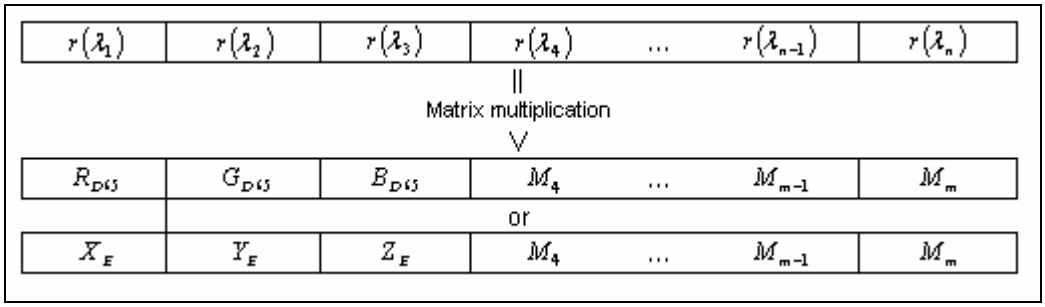

Figure 3.1 Examples for the spectral high definition color description (SCD). The data format is compatible to a conventional tristimulus description and performs device and document-independent high definition color description by encoding the reflectance spectrum of each pixel of the document.

The performance of the SCD can be improved significantly if only a subset of the metameric components M_i is used. Of course, such an encoding scheme is not longer lossless. The trick is to find the components M_i that can be dropped without causing large color errors.

Unfortunately, the influence of a metameric component M_i on the perceived color is dependent on the illuminant. For example, for the SCD basis illuminant, none of the metameric components has any influence.

Several studies showed, that natural reflectance spectra can be described using a small number of eigenvectors. For example, a study by Parkkinen et al. showed that 8 eigenvectors are sufficient to describe the reflectance vectors of 1257 different Munsell colors with minimal errors [21].

Similarly, a limited number of metameric components in addition to the visible components are sufficient to describe a reflectance spectrum with small errors. And indeed, Keusen and Praefcke [21] were able to achieve excellent results. The number of components necessary for a given illuminant was dependent on the similarity between this illuminant and the SCD basis illuminant (D65). Seven components were sufficient for most of the examined illuminants (C, D50, E, A) to achieve maximum color errors below $3\,\Delta E_{ab}$, with mean errors of about 0.2 ΔE_{ab},. The illuminant F2 was the only exception. Here, 10 components were necessary for the same accuracy. The bad performance of the SCD for the fluorescent illuminant is easy to explain, as slight errors of the reflectance spectra at the peak wavelength get amplified.

König and Praefcke complemented this work with an examination of the bit rates necessary for each channel [5], [15], [21], [22], [23], [24], [39]. They used a slightly different set of illuminants (D65, 3200 K, Fl1 and E). To achieve results comparable in quality to 3*8 bit for

all illuminants, 46 bits were necessary. To obtain a quality comparable to 3*12 bit RGB, 79 bits had to be used. This is equivalent to data enlargement factors of 1.9 and 2.2, respectively.

Keusen's method was based on an orthogonal description of the metameric subspace. If this constraint is dropped arid non-orthogonal analysis-synthesis vectors are used for the description of the metameric subspace, further improvements of the performance of the SCD are possible.

A yet unpublished approach to such an improvement is based on a paper by Viggiano [54] that discusses the sensitivity of the human eye to small changes of the spectral stimuli for different wavelengths. It is easy to understand that the human eye is much less sensitive to small changes of the spectral stimulus at a wavelength where the cone sensitivities almost vanish than to changes near to a maximum of the sensitivity curves. Viggiano derived a function, describing the importance of each wavelength for the perceived color. This function can be used to weight the reflectance spectra of the test set before deriving the principal components, and indeed the performance of the SCD is improved for all examined illuminants (D65, C, Xe, A, Fll). Only 6 components were necessary to achieve a quality comparable to Keusen's method at 7 components and mean errors of about 0.2 ΔE_{94} were encountered. The resulting bit rates have not yet been examined, but data enlargement factors of less than 2 are expected.

The drawback of this method is an increase of the spectral error that might cause large errors in theory. Fortunately, this is rather unlikely in practice, because technical illuminants are optimized with regard to the human eye.

Yet another possibility to construct non-orthogonal analysis-synthesis sets follows the one-mode-analysis (OMA) by Marimont and Wandell [31]. The method is based on the singular value decomposition to minimize the mean squared error in a specially chosen color space and is described in detail in [44]. If this method is applied to the metameric subspace, results comparable to the aforementioned results can be achieved using only 5 components corresponding to a data enlargement factor of about 1.7.

In contrast to the approaches discussed previously, the OMA approach uses a set of illuminants to optimize the analysis and synthesis vector sets. Although a decrease in robustness for illuminants not contained in the optimization set was expected this was not observed.

A different approach to compression is possible, if the reproduction is restricted to a small number of illuminants and linear combinations thereof. This requires the absence or neglectability of mutual illumination. In other words reflections between different scene elements are neglectable. Although this is a restriction, in practice it is usually not such a serious limitation.

Such a scenario allows the use of a DPCM (Differential Pulse Code Modulation) coding. First the tristimulus values for one of the illuminants are transmitted. For each subsequent illuminant only the differences between the tristimulus values under this illuminant and predicted tristimulus values are encoded.

A number of different prediction algorithms were examined by Praefcke and Simon [42], [43]. The simplest prediction algorithm is to use the tristimulus values for the first illuminant as prediction. More elaborate methods include a prediction using a white calibration or the use of the Wiener inverse for a spectral prediction. The last method offers the possibility to use all of the previously transmitted illuminants for the prediction.

It was demonstrated that only 38.8 bits were necessary to encode the set described in [11] for the illuminants D65, C, Xe, A and F11 with mean errors of about 0.2 ΔE_{94} and maximum errors below 1.5 ΔE_{94}. This corresponds to a data enlargement factor of only 1.62.

3.2 Proposed compression method

This section explains methods and algorithms, which was proposed for compression spectral images during this research. Human visual system is much less sensitive to recognize little changes in components of reflectance. That fact allows to apply space transformation and "forget about" the least correlated components of the reflectance vectors.

Principal Component Analysis (PCA) or Karhunen-Loéve transformation was used as the main method of compression. It is one of the famous methods for reducing the dimension of the input vectors. That feature allows PCA to reduce amount of input variables using vectors projection to the new, low dimension coordinate system. More detail about PCA-transformation method is written at the next chapter.

In a case of simple PCA application for the spectral images compression ratio will be proportional to dimension of PCA–transformation. New picture size equal:

$$size_{new} = \frac{k}{N} * size_{original}$$

where N - number of wavelengths (e.g. 81 in our experimental case), and k - number of first eigenvectors, which was used in the PCA-transformation.

Later, application of the PCA for the spectral image data was upgraded by using color segmentation. Clustering is for image's separation to the regions with similar color. The separation allows to use less number of principal components with the similar Root Mean Square Error (RMSE). That improves compression ratio and quality of compressed image.

K-means is one of the simplest clustering methods. It is based on iteration movement of initial cluster's centers. But, unfortunately, initial state is random in this algorithm, and clustering results can be different.

For the result's improvement of PCA application, new method was proposed. This method allows to find clusters distributed along lines. But it is also the mentioned above drawback of k-means.

Proposed algorithm of PCA application with clustering consists of the follow steps:

Clustering method is applied to separate image in respect of colors.

Spectral image appears as the input value. Reflectance vectors have to be prepared for algorithm's functioning. $r_\lambda(x, y)$ transforms to $r_\lambda(t, i)$, where i is number of cluster.

Every cluster data, which appears after clustering, is transformed by PCA.

3.3 Principal Component Analysis

Principal Component Analysis is a data transformation technique. Usually it is used when results of the experiences is not one number, but vector. Principal Component Analysis consists in sequential search of factors (Principal Components).PCA is based on the statistical representation of a random variable. Suppose there is a random vector population x, where

$$x = (x_1, \ldots, x_n)^T$$

and the mean of that population is denoted by

$$\mu_x = E\{x\}$$

and the covariance matrix of the same data set is

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\}$$

The components of C_x, denoted by c_{ij}, represent the covariances between the random variable components x_i and x_j. The component c_{ii} is the variance of the component x_i. The variance of a component indicates the spread of the component values around its mean value. If two components x_i and x_j of the data are uncorrelated, their covariance is zero $(c_{ij} = c_{ji} = 0)$. The covariance matrix is, by definition, always symmetric.

From a sample of vectors $x_1 \ldots x_M$, we can calculate the sample mean and the sample covariance matrix as the estimates of the mean and the covariance matrix.

From a symmetric matrix such as the covariance matrix, we can calculate an orthogonal basis by finding its eigenvalues and eigenvectors. The eigenvectors e_i and the corresponding eigenvalues λ_i are the solutions of the equation

$$C_x e_i = \lambda_i e_i, i = 1, \ldots, n$$

For simplicity it is assumed that the λ_i are distinct. These values can be found, for example, by finding the solutions of the characteristic equation

$$|C_x - \lambda I| = 0$$

where the I is the identity matrix having the same order than C_x and the |.| denotes the determinant of the matrix. If the data vector has *n* components, the characteristic equation becomes of order *n*. This is easy to solve only if *n* is small. Solving eigenvalues and corresponding eigenvectors is a non-trivial task, and many methods exist. One way to solve the eigenvalue problem is to use a neural solution to the problem [38]. The data is fed as the input, and the network converges to the wanted solution.

By ordering the eigenvectors in the order of descending eigenvalues (largest first), one can create an ordered orthogonal basis with the first eigenvector having the direction of largest variance of the data. In this way, we can find directions in which the data set has the most significant distribution

Suppose one has a data set of which the sample mean and the covariance matrix have been calculated. Let A be a matrix consisting of eigenvectors of the covariance matrix as the row vectors.

By transforming a data vector x, we get

$$y = A(x - \mu_x)$$

which is a point in the orthogonal coordinate system defined by the eigenvectors. Components of y can be seen as the coordinates in the orthogonal base. We can reconstruct the original data vector x from y by

$$x = A^T y + \mu_x$$

using the property of an orthogonal matrix $A^{-1} = A^T$. The A^T is the transpose of a matrix A. The original vector x was projected on the coordinate axes defined by the orthogonal basis. The original vector was then reconstructed by a linear combination of the orthogonal basis vectors.

Instead of using all the eigenvectors of the covariance matrix, we may represent the data in terms of only a few basis vectors of the orthogonal basis. If it is denoted the matrix having the K first eigenvectors as rows by A_k, we can create a similar transformation as seen above

$$y = A_K (x - \mu_x)$$

and

$$x = A_K^T y + \mu_x$$

This means that the original data vector is projected on the coordinate axes having the dimension K and transforming the vector back by a linear combination of the basis vectors. This minimizes the mean-square error between the data and this representation with given number of eigenvectors.

If the data is concentrated in a linear subspace, this provides a way to compress data without losing much information and simplifying the representation. By picking the eigenvectors having the largest eigenvalues information is lost as little as possible in the mean-square sense. One can e.g. choose a fixed number of eigenvectors and their respective eigenvalues and get a consistent representation, or abstraction of the data. This preserves a varying amount of information of the original data. Alternatively, we can choose approximately the same amount of information and a varying amount of eigenvectors and their respective eigenvalues. This would in turn give approximately consistent amount of information in the expense of varying representations with regard to the dimension of the subspace.

We are here faced with contradictory goals: On one hand, we should simplify the problem by reducing the dimension of the representation. On the other hand we want to preserve as much as possible of the original information content. PCA offers a convenient way to control the trade-off between loosing information and simplifying the problem at hand.

As it will be noted later, it may be possible to create piecewise linear models by dividing the input data to smaller regions and fitting linear models locally to the data.

Now, consider a small example showing the characteristics of the eigenvectors. Some artificial data has been generated, which is illustrated in the Figure 3.2:

3.3.1 Example

As the input data we suppose vector set of the two dimension vectors $X = \{x_i\}_{i=1}^{17}$.

$x_1 = (0.2,\ 0.3)$, $x_2 = (0.3,\ 0.3)$, $x_3 = (0.3,\ 0.4)$, $x_4 = (0.3,\ 0.5)$, $x_5 = (0.4,\ 0.3)$,

$x_6 = (0.4,\ 0.4)$, $x_7 = (0.4,\ 0.5)$, $x_8 = (0.4,\ 0.6)$, $x_9 = (0.5,\ 0.3)$, $x_{10} = (0.5,\ 0.4)$,

$x_{11} = (0.5,\ 0.5)$, $x_{12} = (0.6,\ 0.3)$, $x_{13} = (0.6,\ 0.4)$, $x_{14} = (0.6,\ 0.5)$, $x_{15} = (0.6,\ 0.6)$,

$x_{16} = (0.7,\ 0.5)$, $x_{17} = (0.7,\ 0.6)$. Considering the method; $\bar{x} = \begin{pmatrix} 0.47059 \\ 0.43529 \end{pmatrix}$ is mean x value marked off as red spot in the figure. Matrix $A^T = \begin{pmatrix} -0.89104 & 0.45392 \\ -0.45392 & -0.89104 \end{pmatrix}$ is the matrix consisting of sorted eigenvectors of the covariance matrix as the row vectors. The vector $\begin{pmatrix} -0.89104 \\ -0.45392 \end{pmatrix}$ is the most principal, because is corresponded the most eigenvalue. It is placed first in matrix A. X has the most significant distribution in this direction.

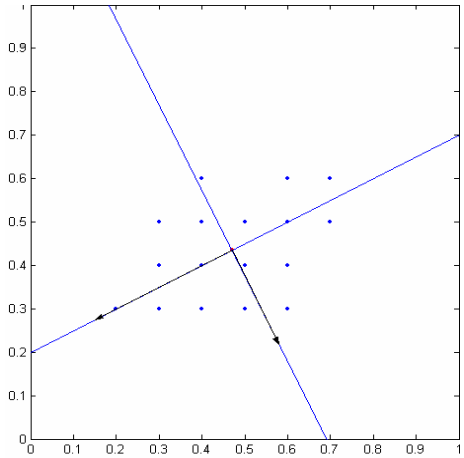

Figure 3.2: Eigenvectors of the artificially created data

3.4 Clustering method

Given a data set $X = \{x_1, x_2, \ldots, x_m\}$ we assume the existence of k clusters whose centers are initially approximated by $y_1^{(0)}, y_2^{(0)}, \ldots y_c^{(0)}$. The process of finding the final values of the cluster centers is iterative. At each step all the patterns are classified and each center is adjusted using a minimizing scheme of an associated performance index which replaces the cluster center by the arithmetic mean of the cluster's samples. The process terminates when there is no difference between two consecutive iterations.

3.4.1 Algorithm

A k-means is an iterative procedure. Given samples in R^n which presumably group around k clusters and initial approximations to the cluster centers, this algorithm calculates the centers iteratively, minimizing at the each iteration a set of performance indices.

Input: n - the vectors' dimension.

m - the number of samples.

k - the number of clusters.

$X = \{x_i\}_{i-1}^{m}, 1 \le i \le m$ - the given samples in R^n.

N - maximum number of iterations allowed.

Output: $\{y_j\}, 1 \le j \le k$ - the final cluster centers.

$\{m_j\}, 1 \le j \le k$ - the cluster sizes.

$\{l_{ij}\}, 1 \le j \le m_j$ - the indices of the original samples which belong to the j cluster, $1 \le j \le k$.

it - the number of iterations needed for convergence.

Step 1. Initialization: set $y_{j0} = x_j, \quad 1 \le j \le k$ and $it = 0$.

Step 2. Classify $\{x_i\}_{i=1}^{m}$, about the cluster centers $\{y_{j0}\}_{j=1}^{k}$ using the minimum distance classifier. For denote $1 \le j \le k$ by $\{x_{l_{ij}}\}_{i-1}^{m_j}$ the samples which cluster around y_{j0}.

Step 3. For $1 \le j \le c$ obtain y_j which minimizes the performance index

$$\mathrm{I}_j(z) = \sum_{i=1}^{m_j} \left\| z - x_{l_{ij}} \right\|^2, \; z \in R^n$$

Basic calculus implies

$$y_j = \left(\sum_{i=1}^{m_j} x_{l_{ij}} \right)$$

i.e. y_j is the arithmetic mean of $\left\{ x_{l_{ij}} \right\}_{i=1}^{m_j}$. Set $it \leftarrow it + 1$.

Step 4. If

$$y_j = y_{j0}, \quad 1 \le j \le k$$

output $y_j, m_j, \left\{ x_{l_{ij}} \right\}_{i=1}^{m_j}, \quad 1 \le j \le k; it$ and stop. Otherwise, if $it \ge N$ output "number of iterations exceeded"; else set $y_{j0} = y_j$ and go to Step 2.

A general optimal choice of k and y_{j0} as well as practical sufficient conditions for convergence of this algorithm are not known, except in obvious cases such as samples that are spread among disjoint cells which are sufficiently apart from each other.

3.4.2 Example

Considering the samples in Fig. 3.3. Assume, it is required two clusters, $k = 2$. Given data state is $X = \left\{ x_i \right\}_{i=1}^{17}$. $x_1 = (0.2,\ 0.3)$, $x_2 = (0.3,\ 0.3)$, $x_3 = (0.3,\ 0.4)$, $x_4 = (0.3,\ 0.5)$, $x_5 = (0.4,\ 0.3)$, $x_6 = (0.4,\ 0.4)$, $x_7 = (0.4,\ 0.5)$, $x_8 = (0.4,\ 0.6)$, $x_9 = (0.5,\ 0.3)$, $x_{10} = (0.5,\ 0.4)$, $x_{11} = (0.5,\ 0.5)$, $x_{12} = (0.6,\ 0.3)$, $x_{13} = (0.6,\ 0.4)$, $x_{14} = (0.6,\ 0.5)$, $x_{15} = (0.6,\ 0.6)$, $x_{16} = (0.7,\ 0.5)$, $x_{17} = (0.7,\ 0.6)$. It is the same as example of PCA. The initial approximations for the two cluster centers $y_{10} = (0.30,\ 0.38)^T$, and $y_{20} = (0.50\ \ 0.36)^T$ were found randomly. y_{10} and y_{20} are marked in the figure as green spot. The samples in the first cluster are $x_1 - x_4, x_6 - x_8$ while $x_5, x_9 - x_{17}$ belong to the second one. The first iteration provides the red cluster's centers: $y_{11} = (0.33,\ 0.43)^T$, $y_{21} = (0.57,\ 0.44)^T$. Since convergence is not obtained, the samples are again classified with respect to the new cluster centers. Now $x_1 - x_8$ are in the first cluster and $x_9 - x_{17}$ are in the second. The second iteration doesn't

give in principal new clusterization result. $y_{12} = (0.34,\ 0.41)^T$, $y_{22} = (0.59,\ 0.46)^T$. Cluster's belonging is $x_1 - x_8$ for first center and $x_9 - x_{17}$ for the second also.

All process of clustering take three iterations in this case. At the last iteration method obtained the same results as at the second and was stopped. The last clustering centers are marked in the figure as blue spots with the black borders.

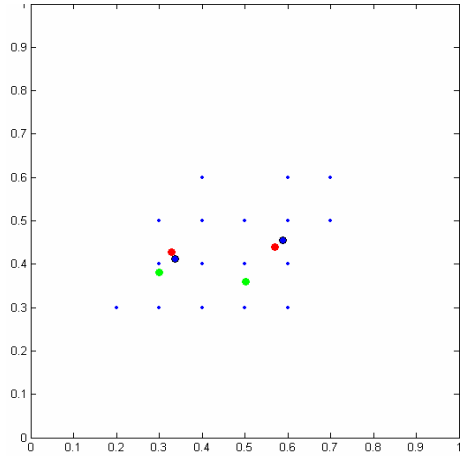

Figure 3.3. Applying k- means clustering algorithm, $k = 2$.

3.5 Adaptation of clustering method for PCA

More suitable clustering algorithm for PCA application was proposed by Jussi Parkkinen [40]. Method based on definition of distance between subspaces and point. Distance of a point x from a subspace M in R^n is $\delta(x, M) = \| x - \hat{x}_M \|$; where $\hat{x}_M$ is the orthogonal projection of x on M and $\| . \|$ is Euclidean vector norm. In future this clustering method will be called as Subspace clustering

3.5.1 Algorithm

Given samples in R^n which presumably group along k clusters (subspaces $L_1, ..., L_K$) and initial approximations to the cluster centers. This algorithm calculates the centers iteratively, minimizing at the each iteration a set of performance indices.

Input: n - the vectors' dimension.

m - the number of samples.

k - the number of clusters.

$L_1, ..., L_K$ - the initial subspaces with given dimensions in R^n.

N - maximum number of iterations allowed.

Output: $\{L_j\}, 1 \le j \le k$ - the final cluster centers.

$\{C_j\}, 1 \le j \le m_j$ - the indices of the original samples which belong to the j cluster, $1 \le j \le k$.

it - the number of iterations needed for convergence.

Step 1. Initialization: set $L_{j0} = x_{random}$, $1 \le j \le k$ and $it = 0$. $L_1, ..., L_K$ is the set of lines, $L_j = L_j^1$, $1 \le j \le k$. It was made this decision because our main aim was the lowest (equal one) dimension transformation in PCA for maximal increasing compression ratio. Clustering method has to find such cluster that is the most suitable for projection process to the one dimension space (i.e. line).

Step 2. Classify all vectors $\{x_i\}_{i=1}^m$, according to the rule $x \in C_j$ if $\| \hat{x}_j \| > \| \hat{x}_r \|$, $r \ne j$, with $\hat{x}_j$ denoting the projection of x on L_j. Settle ties arbitrarily.

As can be known from Higher Mathematics Course [3] that length of projection $\|\hat{x}\|$ of vector $\hat{x}$ to the line (based on vector L) equal Scalar Product of vector and basis vector of line and then divide it to Euclidean vector norm of line basis vector:

$$\| \hat{x}_r \| = \frac{\hat{x}_r * L_j}{\| L_j \|}$$

Step 3. If no changes in classification occur, exit.

Step 4. Compute sample correlation matrix for all clusters.

Step 5. Compute as many eigenvectors belonging to the largest eigenvalues as given by the fixed dimensions (one) and set these as the new basis vectors for the subspaces L_j, $1 \le j \le k$.

Step 6. Go to Step 2.

The same as k-means, Subspace clustering method united input dots to the clusters. But Subspace method's clusters are situated along lines $L_1, ..., L_K$.

3.5.2 Example

Considering the samples in Fig. 3.4. Assume, it is required two clusters, $k = 2$. Given data state is the same as previous examples. $X = \{x_i\}_{i=1}^{17}$: $x_1 = (0.2,\ 0.3)$, $x_2 = (0.3,\ 0.3)$,

$x_3 = (0.3,\ 0.4)$, $x_4 = (0.3,\ 0.5)$, $x_5 = (0.4,\ 0.3)$, $x_6 = (0.4,\ 0.4)$, $x_7 = (0.4,\ 0.5)$, $x_8 = (0.4,\ 0.6)$, $x_9 = (0.5,\ 0.3)$, $x_{10} = (0.5,\ 0.4)$, $x_{11} = (0.5,\ 0.5)$, $x_{12} = (0.6,\ 0.3)$, $x_{13} = (0.6,\ 0.4)$, $x_{14} = (0.6,\ 0.5)$, $x_{15} = (0.6,\ 0.6)$, $x_{16} = (0.7,\ 0.5)$, $x_{17} = (0.7,\ 0.6)$. The initial approximations for the two cluster centers are $L_{10} = (0.71,\ 0.71)^T$, and $y_{20} = (0.89,\ 0.45)^T$ marked in the figure by green spot. L_{10} and L_{20} are random and normalized vectors. The samples in the first cluster are $x_1 - x_8, x_{10}, x_{11}, x_{14}, x_{15}, x_{17}$ while $x_9, x_{12}, x_{13}, x_{16}$ belong to the second one. The first iteration provides the black cluster's centers: $L_{11} = (0.69,\ 0.72)^T$, $L_{21} = (-0.84,\ -0.54)^T$. Now $x_1 - x_4, x_6 - x_8, x_{11}, x_{14}, x_{15}, x_{17}$ are in the first cluster and $x_5, x_9, x_{10}, x_{12}, x_{13}, x_{16}$ are in the second. The second iteration gives in new clusterization result. Here $L_{12} = (0.68,\ 0.73)^T$, $L_{22} = (-0.83,\ -0.56)^T$ are marked by orange spot. Cluster's belonging is $x_1 - x_4, x_6 - x_8, x_{11}, x_{15}, x_{17}$ for first center and $x_5, x_9, x_{10}, x_{12} - x_{14}, x_{17}$ for the second also. The third iteration obtains the final values: $L_{13} = (0.64,\ 0.77)^T$, and $L_{23} = (-0.83,\ -0.56)^T$. Finally, $x_1 - x_4, x_6 - x_8, x_{11}, x_{14}, x_{15}$ consist in the first cluster, and $x_5, x_9, x_{10}, x_{12}, x_{13}, x_{16}, x_{17}$ consist in the second one.

All process of clustering take four iterations in this case. At the last iteration method obtained the same results as at the second and was stopped. The last clustering centers are marked in the figure by blue spots.

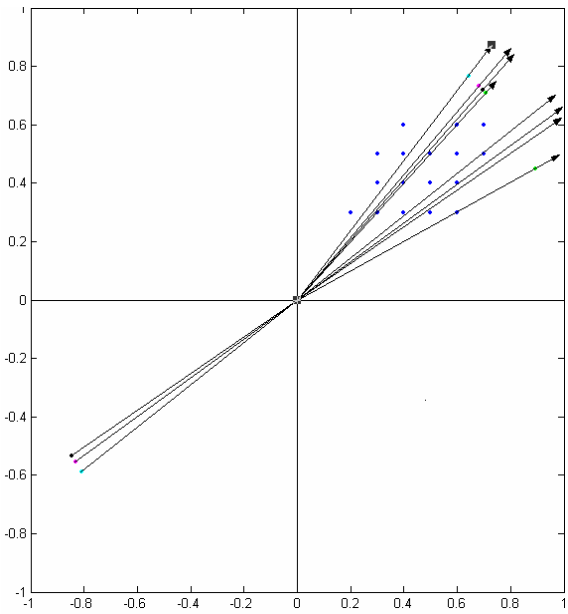

Figure 3.4. Applying Subspace clustering algorithm with $k = 2$.

4 EXPIREMENTS AND RESULTS

4.1 Definitions

This chapter is directed to help for readers to understand meaning of some values, definitions and symbols in the current thesis.

Color Difference - average ΔE -value for the whole picture as.

$$\Delta E = \frac{\sum_{j=1}^{M} \sqrt{\left(L_{j1}^{*} - L_{j2}^{*}\right)^2 + \left(a_{j1}^{*} - a_{j2}^{*}\right)^2 + \left(b_{j1}^{*} - b_{j2}^{*}\right)^2}}{M}$$

Compression Ratio is a value that defines relation between size of compressed file and the original image.

$$ratio = \frac{file\ size_{original\ image}}{file\ size_{compressed\ image}}$$

Root Mean Square Error (RMSE) is a characteristic that shows average Euclidean distance between original and transformed vectors.

$$RMSE = \frac{\sum_{j=1}^{M} \sqrt{\sum_{i=1}^{N} \left(x_{ij} - y_{ij}\right)^2}}{M}$$

where, N - dimension of the vectors space. M - number of vectors.

Loss means difference between original and compressed image. It can be Root Mean Square Error or Color Difference.

Dimension of (PCA) transformation is a number of the eigenvectors corresponding largest eigenvalues of the covariance matrix used in PCA-transform.

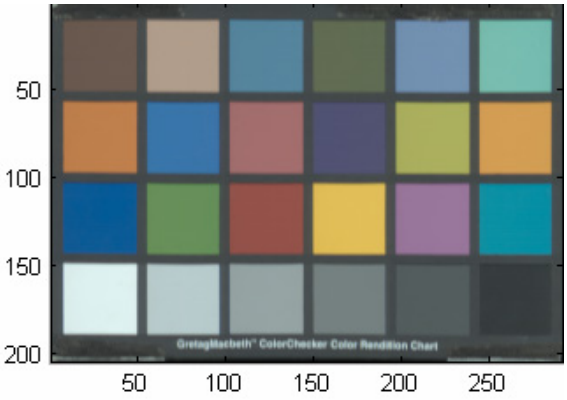

Figure 4.1. RGB representation of Colorchecker image. Image size is 204×291 pixels, 81 channels.

Figure 4.2. RGB representation of Icon image. Image size is 394×321 pixels, 81 channels.

The platform in experiments was PC with 1.8 GHz CPU and 2 GByte RAM. The algorithms were coded in Matlab 6.2. A representation of the tested images is shown in Figure 4.1 and Figure 4.2.

4.2 Simple PCA-transformation application

At the first step of the research Principal Component Analysis was applied to the Spectral Image data. PCA was decided to employ for spectral image data because the fields of application of PCA are connected with high-dimension data (e.g. clustering [1] or pattern recognition [55]). Originally, PCA allows to reduce dimension of the input vectors and this was most suitable for our problem decision. But unfortunately, losses appear during transformation. The main aim of this research was to estimate compression dependency from

dimension of compressed image and compression ratio. For estimating of compression losses, two values - ΔE and Root Mean Square Error were chosen. RMSE is a standard value to estimate compression losses. It represents mean square distance between original and compressed data. But for estimation visual losses, which are appeared, RMSE is not enough. ΔE is the value that represents difference respecting to the human visual system, distance in CIE $L^*a^*b^*$ space.

Looking at the results (Table 4.2), RMSE decreases as dimension increases. The change is smooth and looks like the function $\frac{1}{x}$.

If in compression the dimension of the transformed image is half of number of wavelengths, the result will be very similar to original. You can see, that $\Delta E_1 = 22.32$ (Color Difference when dimension is equal 1), $\Delta E_2 = 10.642$. From this it follows that $\Delta E_1 - \Delta E_2 = 11.68$ - very big difference. But $\Delta E_9 - \Delta E_{10} = 0.069$ ($\Delta E_9 = 0.29$). It's mean almost the same. This all facts narrow interval of searching optimal compression parameters.

Figure 4.3. One and five dimension transformed image

This can be seen in Figure 4.3. Left image is the result then only first eigenvector in spectral compression is used. It looks like gray-scale image and color difference ΔE to original image is 22.32. Right figure shows 5-dimension PCA transforms, $\Delta E_5 = 1.31$. Difference to the original image is almost not visible.

Compression ratio is almost linear value. It depends on Dimension of the Transformation.

$$Ratio = \frac{size_{original}}{d/D * size_{original} + d * D} \approx \frac{D}{\mathrm{d}}$$

where, d is dimension of the transformation. D - number of wavelengths. The above approximation is valid for practically used values, when large image and high compression.

After transform, the original image is represented by two components - reduced color vectors which size equals dimension of the transformation and transformation matrix A. A consists of eigenvectors of the covariance matrix, which has a constant size equaling number of wavelengths (81 in our experiment); number of these vectors equals d.

The Old Orthodox Icon was used in experiments as the main object of the research (Figure 4.2). There are three main colors in the icon image: green, gold and red. The background is white. Inscriptions and contours are black. In such a way, icon has five main colors. That is the reason why increasing the dimension of the transformation after five improves ΔE not so quickly as before five.

Table 4.1. Dimension of the transformation, Compression Ratio and Color Difference ΔE

Dimension	1	2	3	4	5	6	7	8	9	10	40	81
Ratio	81	40,5	27	20,3	16,2	13,5	11,5	10,1	9,0	8,1	2,0	1
ΔE	22,31	10,64	7,07	2,60	1,30	1,22	0,70	0,29	0,29	0,22	0,07	0

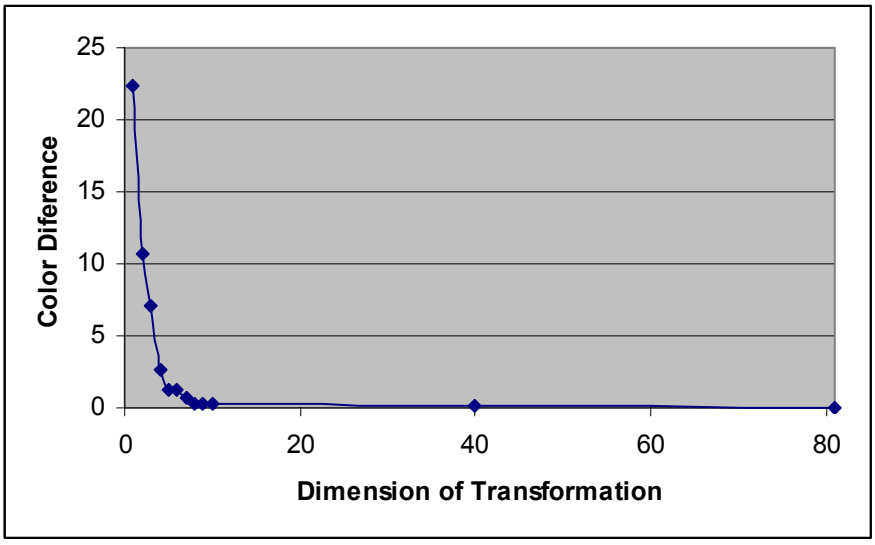

Figure 4.4. ΔE dependency from Compression Ratio

Table 4.2. Dimension of the transformation, Compression Ratio and Root Mean Square Error

Dimension	1	2	3	4	5	6	7	8	9	10	40	81
Ratio	81	40,5	27	20,3	16,2	13,5	11,5	10,1	9,0	8,1	2,0	1
RMSE	0,615	0,165	0,088	0,066	0,051	0,043	0,037	0,033	0,031	0,029	0,0010	0

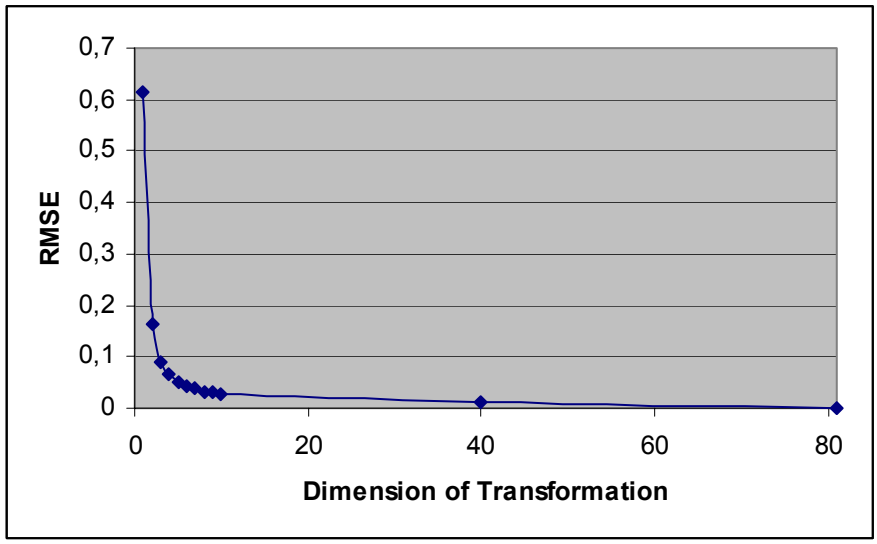

Figure 4.5. RMSE dependency from Compression Ratio.

4.3 Application of PCA with clustering

This chapter explains improvements of the Principal Component Analysis method by adding clustering. It was proposed that losses of the compression will be less if PCA-transform be employed separately for different picture parts with similar vectors. It allows to reduce average distance between new space (space of the principal components) and original vectors. Image was divided (clustered) to the different segments in respect of colors. Then, PCA was applied for the every cluster separately. New data is calculated for every part, and segments are pasted to the new image together. For estimation of the difference between original and compressed image, the RMSE and ΔE were calculated for the whole image data.

It was decided to apply two clustering methods. At the first step of clustering addition, k-means method was chosen for image segmentation. K-means is one of the simplest clustering methods. It allows to separate input data to units as many as needed, with respect of the data closeness. In our case this kind of data is spectral image vectors. But Clustering method can be applied at another color spaces (e.g. $L^*a^*b^*$ or RGB). In this research Spectral Space, $L^*a^*b^*$ and a^*b^* (a^*b^* is the $L^*a^*b^*$ space without luminescent coordinate L^*) ware used.

It is used spectral image data, so the spectral space has to be used in the research. A color is a result or a phenomenon created by the human visual system. So the color is one kind of subjective information. On the other hand, the spectral reflectance is one kind of objective

information, and it is a source of the color. One of the main aims of the current research is to establish more highly accurate compression method in the sense of both objectively and subjectively. Therefore, a suitable color space is also required in the research. CIE $L^*a^*b^*$ is a color space, and it is a uniform color space. In the consideration of clustering, compression and its evaluation, the uniformity is important. Then, CIE $L^*a^*b^*$ color space is chosen. The last one, a^*b^*, is enough and suitable if interest is in hue or chroma of the color. As a result, the spectral, $L^*a^*b^*$, and a^*b^* are used in the research.

The subspace clustering method was applied as a new method of clustering. It allows to find clusters more adopted to the Principal Component Analysis (See 3.5). The method finds clusters as elements distributed along cones in spectral space. Although tip of cone can only go through the point of origin, it was hypothesized to decrease the transformation losses.

As the results of the experiments, 24 tables and diagrams was obtained (See Appendix). For the better comparison of the results, two images were tested – Colorchecker (See Figure 4.1) and Old Orthodox Icon (See Figure 4.2). The size of the images in Matlab format are equal 36.6 MB and 78.1 MB, respectively. Images were tested for 1-20 numbers of clusters and from 1 to 6 dimension of the transformation. Result for the Root Mean Square Error and ΔE of method implementation in order of clustering in Spectral, a^*b^* and $L^*a^*b^*$ spaces by k-means and subspace clustering are shown in Appendix.

Compression ratio in this case differs from simple PCA-transform application. New compressed file format consists of reduced image data, matrix of belongings to clusters and color transformation matrixes. Compression ratio for every case of compression can be calculated using following equation:

$$Ratio = \frac{Size_{original\ file}}{Size_{original\ file} * d/D + Size_{clusters} + k * (D * d)}$$

where $Size_{original\ file}$ - the size of original image, d - dimension of the transformation, k - number of clusters, D - number of wavelengths, and $Size_{clusters}$ - the size of matrix of belongings to clusters. Matrix with the same width and height as original image consists of integer numbers. Each number determines the cluster to which the corresponding pixel belongs. For example, table of some Compression Ratios are shown bellow.

Table 4.3. Compression Ratio for some variations of number of clusters and dimension of the transformation (Colorchecker).

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	71,93	38,09	25,90	19,62	15,79	13,22
4	71,81	38,02	25,85	19,59	15,76	13,19
9	71,60	37,90	25,77	19,52	15,72	13,15
25	70,96	37,54	25,52	19,33	15,56	13,02

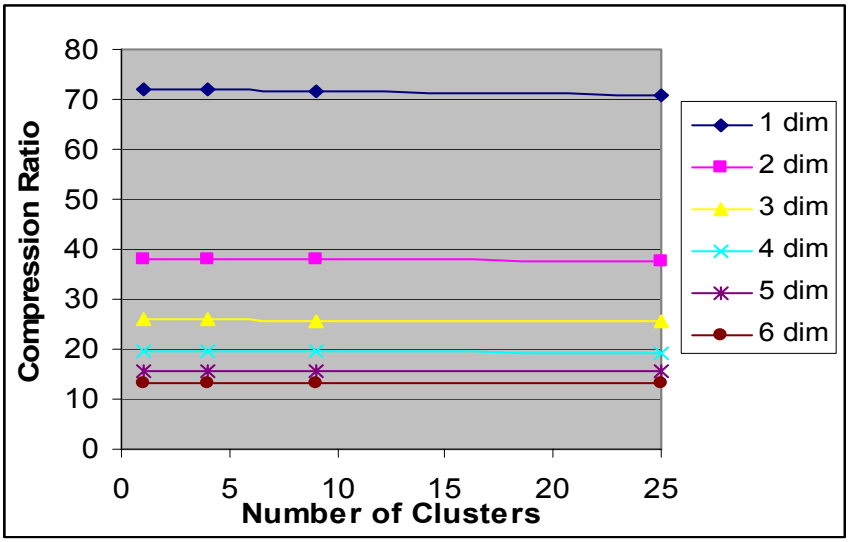

Figure 4.6. Compression Ratio dependency from Number of Clusters.

Table 4.4. Compression Ratio for some variations of number of clusters and dimension of the transformation (Icon).

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	71,85	38,05	25,88	19,61	15,78	13,20
4	71,59	37,91	25,78	19,53	15,72	13,15
9	71,16	37,66	25,61	19,40	15,61	13,07
25	69,82	36,92	25,09	19,00	15,29	12,79

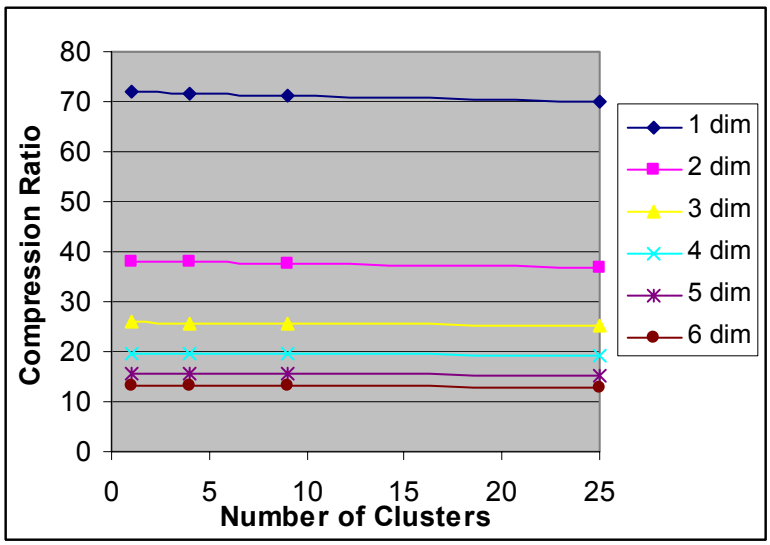

Figure 4.7. Compression Ratio dependency from Number of Clusters.

Considering this formula, that *Ratio* can be less than one then $\left(Size_{original\ file} * {}^{d}\!/_{D} + Size_{clusters} + k * d^2\right) > Size_{original\ file}$. It can happen that dimension of the transformation tends to number of wavelengths, or number of clusters is too high. It is one of the reasons, why there should be limited number of clusters and dimension of the transformation.

Main aim of these tables and diagrams is to show dependency of compression ratio and ΔE on different parameters (i.e. number of clusters, image content and dimension of PCA transformation). As it can be seen, the quality of image increases (ΔE decreases) when the dimension and number of clusters increase. It means that the problem of the optimal solution search for even one picture becomes wider than for simple PCA application.

Dimension of the transformation influences to compression ratio more than number of clusters (See Table 4.3 and Table 4.4). Even for the case of 25 clusters, the compression ratio for 1 dimension is more then case of 1 cluster and 2 dimension space. Table 4.5 and Table 4.6 show (shortly) dependency of the ΔE on the number of clusters for the first principal component. You can see that all graphics starts from one point corresponding to 1 cluster (one cluster means that PCA was applied without clustering). Then, all graphics approximately goes down, but some irregularity takes place. “Jumps” (e.g. clustering in Spectral space using k-means method, in the case of seven clusters clusterization, see Table 4.5 or clustering in $L^*a^*b^*$ space using Subspace clustering method, in the case of four clusters clusterization, see Table 4.6) may appear as the result of random initial state of clustering methods. This undeterminism affects to the algorithm results. ΔE and RMSE increases although number of clusters increases too. It was mentioned about this effect in [50] also.

Table 4.5. Color Difference ΔE of the compressed/decompressed Colorchecker. Two clustering methods applied in the different color spaces.

	$a*b*$ space k-means	$L*a*b*$ space k-means	Spectral space k-means	$a*b*$ space Subspace	$L*a*b*$ space Subspace	Spectral space Subspace
1	23,75	23,75	23,75	23,75	23,75	23,75
4	10,70	10,38	14,09	20,77	11,10	17,88
9	5,32	6,44	8,03	14,78	4,81	6,60
25	1,42	1,73	2,29	7,26	1,83	1,58

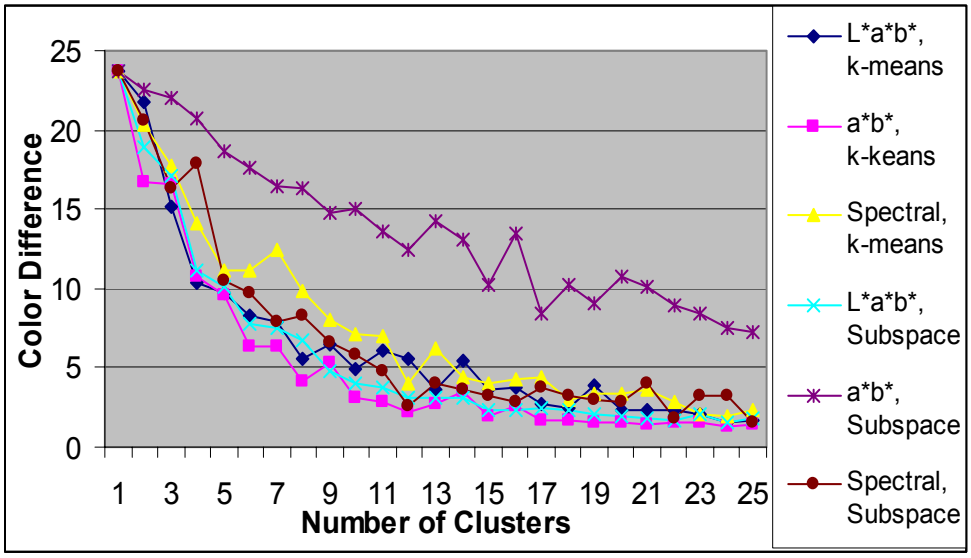

Figure 4.8. Dependency of the Color difference ΔE on the number of clusters for the first principal component. Colorchecker.

Table 4.6. Color difference ΔE of the compressed/decompressed Icon. For two clustering methods applied in the different color spaces.

	$a*b*$ space k-means	$L*a*b*$ space k-means	Spectral space k-means	$a*b*$ space Subspace	$L*a*b*$ space Subspace	Spectral space Subspace
1	18,95	18,95	18,95	18,95	18,95	18,95
4	6,92	6,01	7,87	11,79	5,63	7,13
9	4,44	3,67	4,87	11,16	3,80	3,40
25	3,28	2,47	3,50	10,39	2,79	2,62

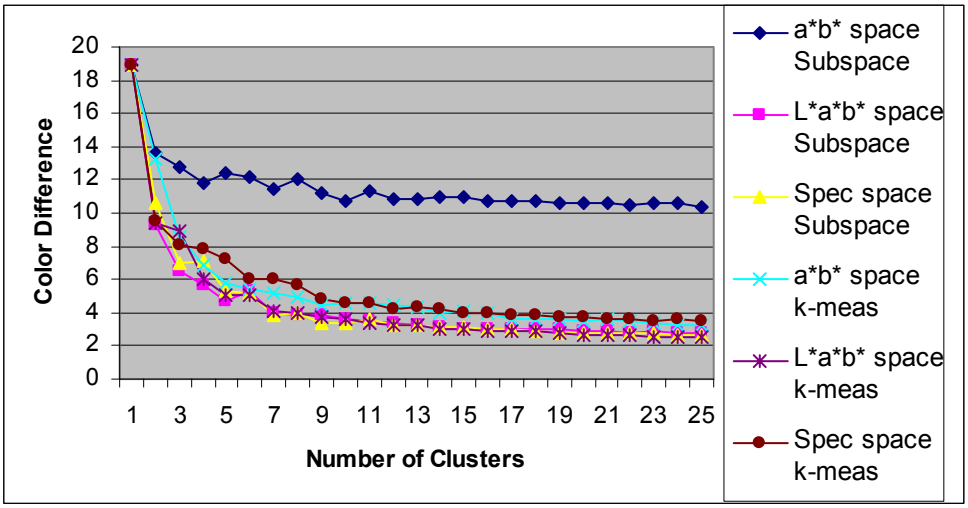

Figure 4.9. Dependency of the color difference ΔE on the number of clusters for the first principal component. Icon.

Considering graphics, the worst case of clustering (as you can see on the Table 4.5 and Table 4.6 for both images) is in $a*b*$ space by Subspace method. Even in the case of clustering image to the 25 parts in respect of $a*b*$ color coordinates, Color Difference is still too high, $\Delta E = 10.40$ for Icon and $\Delta E = 7.26$ for Colorchecker. Figure 4.14 and Figure 4.11 show representation of the compressed images and rough picture of clusters distribution. Image with the same width and height as original image, where each color determines the cluster to which correspond pixel belongs. The application of the k-means method in Spectral spaces for both images doesn't show good results as well.

As the best way, considering the tables, of clustering for proposed compression method is k – means clustering in $a*b*$ space for Colorchecker and $L*a*b*$ for Icon. You can see that Figure 4.10 and Figure 4.13 is almost the same as originals and that distribution of clusters is smooth. Here, Color difference $\Delta E = 1.42$ for Colorchecker and $\Delta E = 2.47$ for Icon.

In addition, representations of the images clustered by Subspace clustering in Spectral space are shown in Figure 4.12 and Figure 4.15. New method showed good results (but not the best) and can be considered as suitable for applying for compression. Compressed Colorchecker

differs from original by $\Delta E = 1.58$ is almost the same as k-means clustering in a^*b^* space. For Icon this value is equal 2.62.

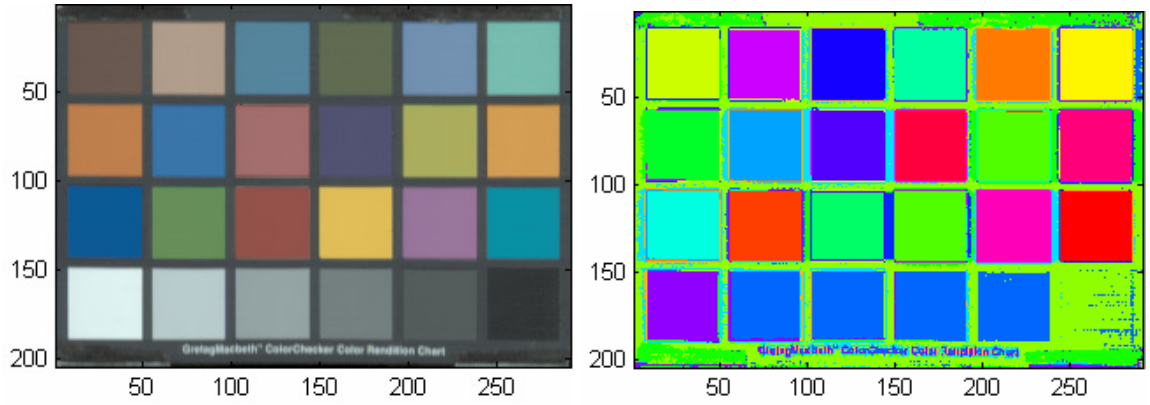

Figure 4.10. Representation of the Colorchecker compressed by using 25 clusters in a^*b^* space (k-means method) and representation of clusters distribution.

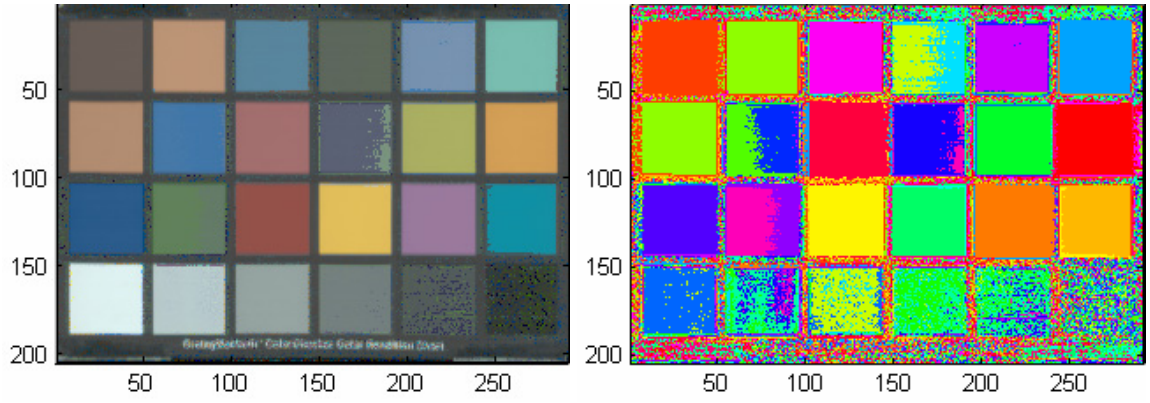

Figure 4.11. Representation of the Colorchecker compressed by using 25 clusters in a^*b^* space (Subspace clustering method) and representation of clusters distribution.

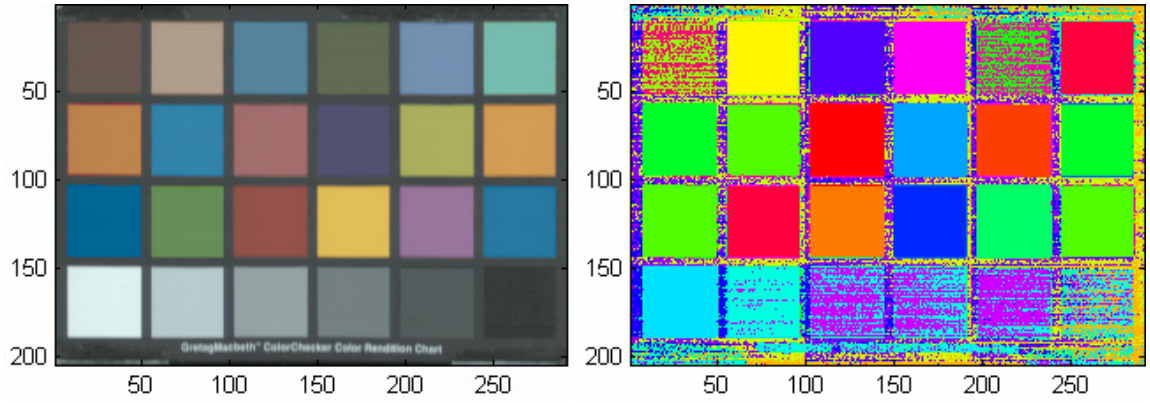

Figure 4.12. Representation of the Colorchecker compressed by using 25 clusters in Spectral space (Subspace clustering method) and representation of clusters distribution.

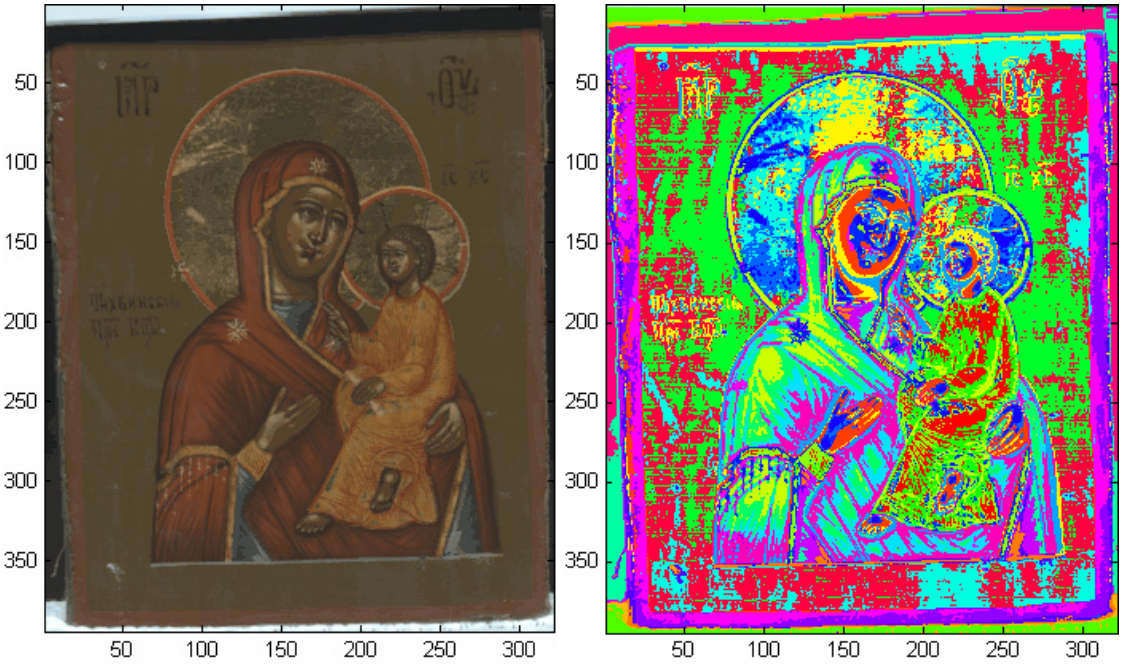

Figure 4.13. Representation of the Icon compressed by using 25 clusters in $L^*a^*b^*$ space (k-means) and representation of clusters distribution.

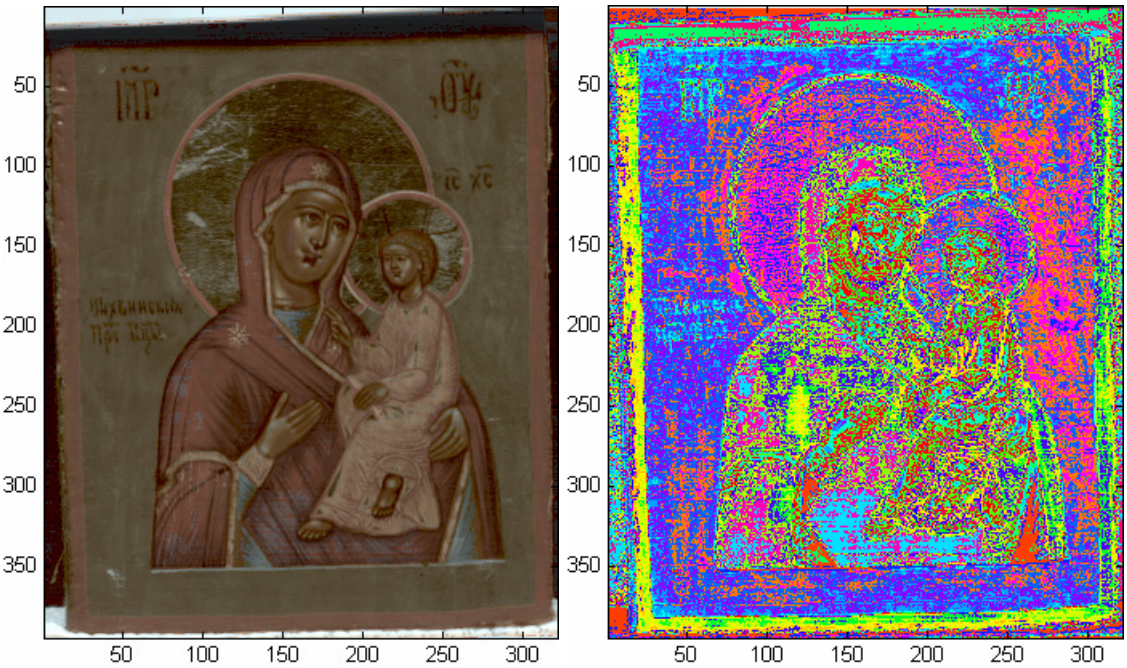

Figure 4.14. Representation of the Icon compressed by using 25 clusters in a^*b^* space (Subspace clustering method) and representation of clusters distribution.

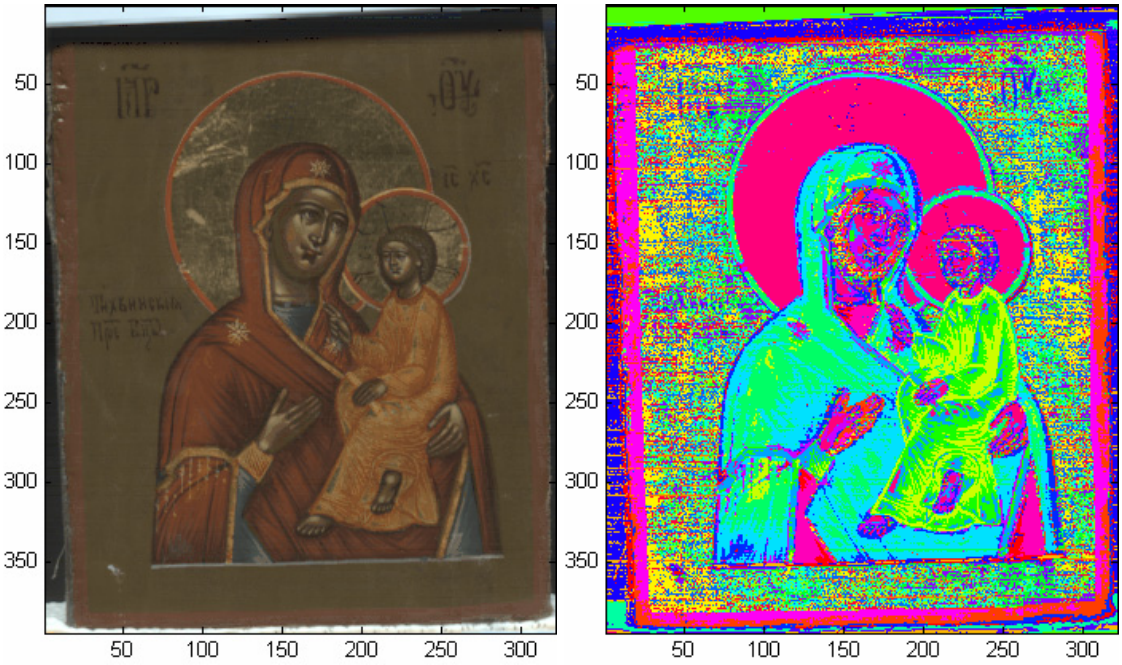

Figure 4.15. Representation of the Icon compressed by using 25 clusters in Spectral space (Subspace clustering method) and representation of clusters distribution.

It is hard to understand whether the subspace clustering has advantage or not. Consider an image (see Figure 4.16) to show advantage. This image (of size 100×100 pixels, 81 channels) consists of linear distributed color vectors. Colors in the same column pixels differ from each other by little random noise ($r_\lambda(y) = r_\lambda(y) + 0.1 * random$). Colors in the same row pixels differ from each other by linear component and little random noise too ($r_\lambda(x) = a * r_\lambda(x-1) + 0.1 * random$). Where: $r_\lambda(y)$ - spectral components of pixels in the column y; $r_\lambda(x)$ - spectral components of pixels in the row x; $random$ - random number from 0 to 1; a - linear component.

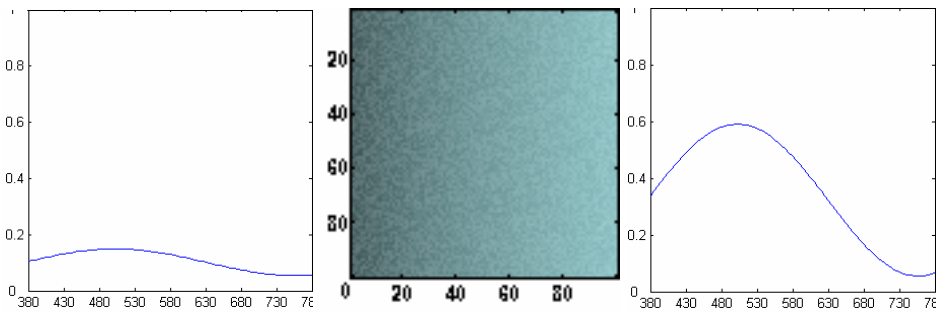

Figure 4.16. Spectral of the lowest color, original image, spectral of the highest color.

Compare color difference ΔE presented by compression with subspace and k-means clustering in spectral space (2 clusters and 1 dimension). Figure 4.17 and Figure 4.18 show

results of clustering and corresponding compression. In the case of k-means clustering color difference ΔE of 1,34, while for subspace clustering, color difference ΔE of 0,76.

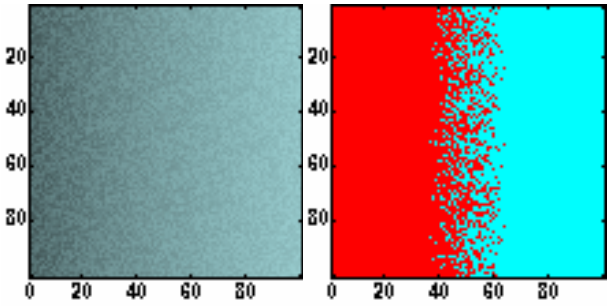

Figure 4.17. Representation of artificial image compressed using k-means clustering.

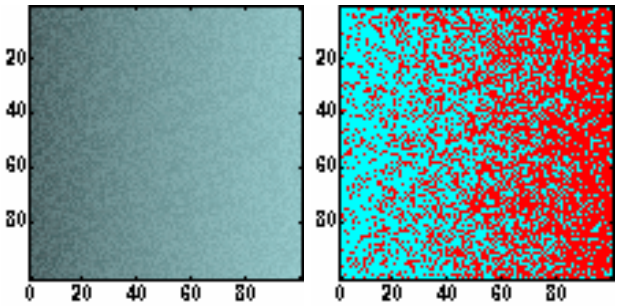

Figure 4.18. Representation of artificial image compressed using subspace clustering.

Considering the results, it can be concluded that subspace clustering obtains better results than conventional methods in a case of linear color distribution.

4.4 Methods comparison

Additionally, all proposed methods was compared with already exists method. For comparison, application JPEG compression method for every spectral image layer was chosen. In the JPEG image compression algorithm, the input image is divided into 8-by-8 blocks, and the two-dimensional DCT is computed for each block. The DCT coefficients are then quantized, coded, and transmitted. The JPEG receiver (or JPEG file reader) decodes the quantized DCT coefficients, computes the inverse two-dimensional DCT of each block, and then puts the blocks back together into a single image. For typical images, many of the DCT coefficients have values close to zero; these coefficients can be discarded without seriously affecting the quality of the reconstructed image.

Number of nonzero DCT coefficients $n \in [1, 64]$ was chosen as the main parameter of JPEG compression. Compression ratio in this case is proportionate to n and equal $\frac{64}{n}$. Results of compression are two tables shown bellow:

Table 4.7. Compression results of JPEG application for spectral image Colorchecker, and corresponding results of the proposed method.

JPEG				PCA with clustering			
n	ΔE	RMSE	Compression Ratio	dim	ΔE	RMSE	Compression Ratio
1	6,49	0,3906	64,00	1	1,42	0,0190	70,96
3	4,18	0,2375	21,33	3	0,76	0,0099	25,52
6	2,94	0,1704	10,67	7	0,38	0,0059	11,05
21	1,05	0,0875	3,04	26	0,09	0,0030	3,00
36	0,37	0,0532	1,78	43	0,05	0,0019	1,80
49	0,23	0,0379	1,31	60	0,02	0,0011	1,30
58	0,16	0,0242	1,10	72	0,01	0,0007	1,10
64	0,09	0,0039	1,00	79	0,002	0,0002	1,01

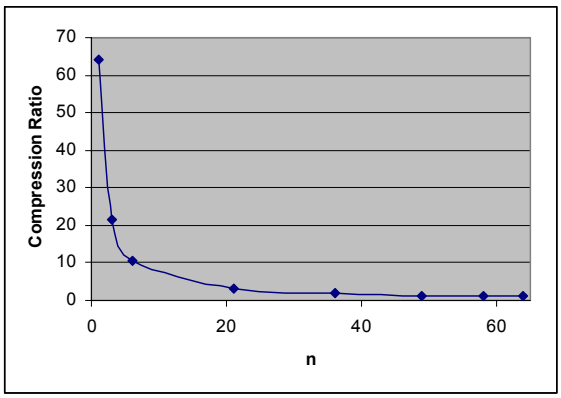

Figure 4.19. Compression Ratio dependency from DCT coefficient.

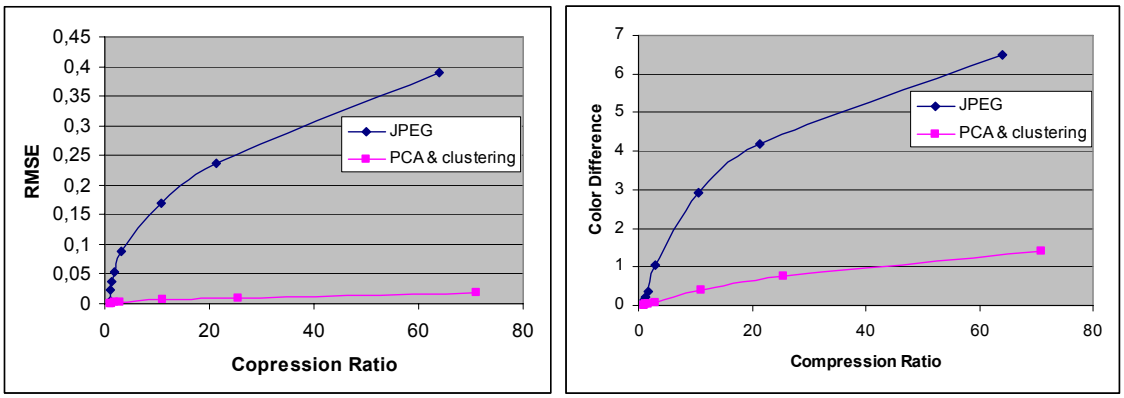

Figure 4.20. RMSE, Color Difference ΔE dependency from Compression Ratio.

Table 4.8. Compression results of JPEG application for spectral image Icon.

JPEG				PCA with clustering				PCA			
n	ΔE	RMSE	Compression Ratio	dim	RMSE	ΔE	Compression Ratio	dim	RMSE	ΔE	Compression Ratio
1	5,62	0,2800	64,00	1	0,0152	2,47	69,82	2	0,1650	10,64	40,5
3	4,26	0,1810	21,33	3	0,0082	1,42	25,09	4	0,0660	2,60	20,20
6	3,20	0,1410	10,67	7	0,0050	0,59	11,05	8	0,0333	0,29	10,10
21	1,59	0,0818	3,04	26	0,0025	0,17	2,99	27	0,0026	0,12	2,99
36	0,88	0,0557	1,78	43	0,0015	0,09	1,84	44	0,0014	0,07	1,84
49	0,53	0,0408	1,31	60	0,0008	0,02	1,32	61	0,0009	0,01	1,32
58	0,32	0,0297	1,10	72	0,0005	0,01	1,10	73	0,0005	0,004	1,10
64	0,12	0,0151	1,00	79	0,0002	0,002	1,01	80	0,0001	0,0004	1,00

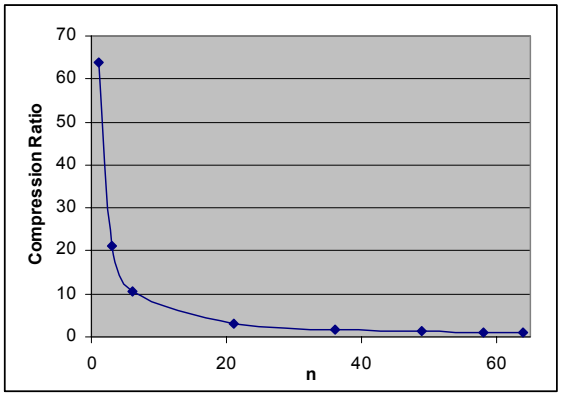

Figure 4.21. Compression Ratio dependency from DCT coefficient.

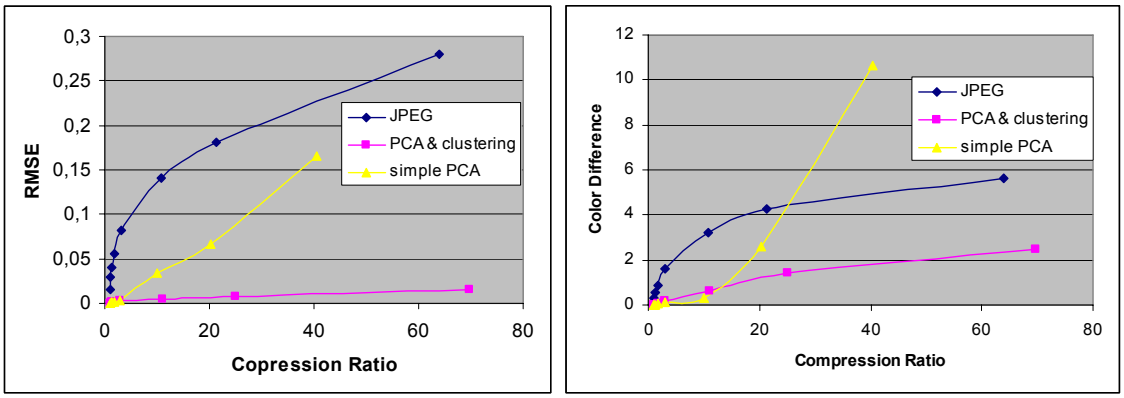

Figure 4.22. RMSE, Color Difference ΔE dependency from Compression Ratio.

Compare compression losses presented by JPEG and proposed methods obtaining the same compression ratio. Considering diagrams (see Figure 4.20 and Figure 7.19), compression ratio for the JPEG corresponds bigger losses then PCA application with the clustering and simple PCA almost everywhere. For example Colorchecker, compression ratio is about value of 10. It correspond JPEG application with $n = 6$ and proposed method compression using 25 clusters and seven dimension transformation. ΔE is equal 2,94 for JPEG application and 0,38 for PCA application with Subspace clustering in the Spectral space.

Comparing losses for Icon, it can be seen that simple PCA application obtains better results until compression ratio is less then 12. It corresponds to six dimension transformation with clustering ($ratio = 12{,}79$, $\Delta E = 0{,}71$) and seven dimension of the simple PCA transformation ($ratio = 11{,}50$, $\Delta E = 0{,}70$). Further, clustering improves ratio and losses.

5 DISCUSSION

This chapter summarizes research, discuses about advantages and disadvantages of proposed methods and hypothesized the most suitable method's application.

In this master's thesis, methods for spectral image compression were proposed. For more quantitative evaluation, compression methods were tested with different parameters.

Two spectral images were used for testing, Macbeth ColorChecker and Old Orthodox Icon. First as a good reflectance image with standard colors and the second one as a typical images for Museum's Archive Systems.

From the results of the first part of the research (PCA application for spectral image data), one can see following:

- It has been determined that application of the Principal Component Analysis for a spectral image reduces amount of data. PCA can be applied as a compression method.
- Increasing the dimension of PCA transformation decreases compression losses but increases the size of compressed file.

As an example, simple PCA transformation can be applied for distributed archive systems. Database stores original spectral images. When user wants to download an image, one dimensioned transformed image is presented as preview. Data size is minimal and quality is low. It allows network quickly transmit preview and user easily chose one, which are useful for him. Then, the user downloads original image or any dimensioned transformed version depending on user wishes.

Addition of a clustering method was proposed as a second step to improve the PCA transformation. It was proposed that generally many pixels with similar colors occur in images with high probability. Therefore, it is possible to reduce redundancy in the image by clustering the image to sets of pixels with a similar color. That image division improves PCA results. Losses decrease for the same dimension of the transformation compared to the PCA. This approach of dividing the spectral image compression into three phases (image clustering, spectrum reduction and encoding) seems to be reasonable. As the result of testing it follows:

- Color Difference ΔE and RMSE for the same dimension of PCA transformation decreases while number of clusters decreases. (See Appendix)

Increase in the number of the clusters increases size of the compressed file not as quick as dimension of PCA transformation. Even in a case of compression with 25 clusters and one dimension transformation, compression ratio is more than one cluster and two or higher dimension of the transformation. (See Table 4.3 Table 4.4)

Application of the clustering methods in different color spaces gives different results for the same image. It can be seen assuming Table 4.5 and Table 4.6, that k-means clustering is preferable applied in $a*b*$ for Colorchecker while for Icon, $L^*a^*b^*$ space clustering is better than $a*b*$.

Subspace clustering gives not the best results only in some cases, but good overall results. Unfortunately, proposition about result's improvement wasn't justified hopes. Compression with Subspace clustering shows almost the same results as simple k-means.

Comparing compression losses presented by JPEG and proposed methods obtaining the same compression ratio, clustering extends compressed file. Considering Figure 4.22, efficacy of simple PCA becomes better for the compression ratio is less 12.

All proposed and tested methods, which were realized during this study, are lossy compression. Our efforts were focused to minimize losses in combination with maximal compression ratio. The method, complicated by clustering, demand high computation efforts and is time consuming. From this, it follows that improvements of PCA can't be applied for real time system. But on the other hand, decompression process doesn't take a lot of time and memory. This is the most suitable for archive systems.

Image can be clustered during time. Then, the best clustering result is obtained. Image is possible to save in database. The file doesn't take much space on the database hard drive, and the user can adjust the quality of stored image.

Of course, proposed methods are not the last step in research process for the spectral image data. First of all, the clustering method can be improved. This is a wide area for research. It is possible to create specific clustering method, which will be adopted for different branches of science. For instance, if old icons are sample for compression, then clustering method has to assume color distribution around black, grey, dirty-gold, dark-green and dark-blue. If image to be compressed is from cosmos, method has to assume that there is a lot of black color in

image and every bright spot is very important. If talk is about medicine, process of recognition is more important in first step of the research. Then image can be divided to important part (e.g. where illness is present) and unimportant one.

Besides clustering method improvement, it is possible to apply any appropriate compression method for compressed image. For instance, compression methods proposed in the current master thesis produce n - layers images in compression format. It can be applied DCT [2] or Huffman [19] for compressed data. It allows to compress data better.

All aims of current research were accomplished.

Principal Component Analysis is successfully applied for the spectral image data.

It was proved that increase of dimension in PCA transformation decrease compression losses (ΔE and RMSE).

The PCA was improved by clustering, that allowed decrease RMSE and ΔE as well.

The subspace clustering method was proposed and tested for the improvement results of PCA transformation.

It was proved that increase in number of clusters decrease compression losses (ΔE and RMSE). Moreover, it was proved that subspace clustering method improves results of compression in the increase of number of clusters, but not as quickly as k-means.

The methods were compared with one of the famous compression method (JPEG). It was proved; that PCA is much more effective method than JPEG or simple application of PCA.

Areas of methods applying were suggested.

Improvements of the method were proposed in discussion part.

Summarizing the studying, the best compression technique is one dimension transformation applied on 25 clusters. K-mean clustering has to be applied in $a*b*$ color space for the Colorchecker and in $L*a*b*$ space for the icon. ΔE achieves in this case bellow 3. It means that human visual system doesn't be able to recognize difference. Compression ratio is about 70. It corresponds to data enlargement factor is about 0,38.

6 REFERENCES

[1] **Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.,** Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, *Proceedings of SIGMOD'98*, Seattle, Washington, June 1998.

[2] **Ahmed, N., Natarajan, T. Rao, K. R.,** Discrete Cosine Transform, *Proceedings of the IEEE Transactions on Computers*. pp. 90-93, Jan. 1974.

[3] **Barber, D.,** Learning from Data 1. Supplementary Mathematics (Vector and Linear Algebra), **http://www.anc.ed.ac.uk/~dbarber/lfd1/lfd1_supp_maths.pdf** , 13th Apr. 2004.

[4] **Bernes, R. S**., Home Webpage **http://www.cis.rit.edu/people/faculty/berns/**, 8th Apr. 2004.

[5] **Berns, R. S.,** Challenges for Color Science in Multimedia Imaging, *Proceedings of CIM'98 Colour Imaging in Multimedia*, Derby, UK, pp. 123-133, Mar. 1998.

[6] **Berns, R. S., Imai, F. H., Burns, P. D., Tzeng, D.-Y**., Multi-Spectral-Based Color Reproduction Research at the Munsell Color Science Laboratory, *Proceedings of Electronic Imaging: Processing, Printing, and Publishing in Color,* Zurich, Switzerland, pp. 106-111, May 1998.

[7] **Burns, P. D.,** *Analysis of Image Noise in Spectral Color Acquisition*, Dissertation at the Rochester Institute of Technology, Rochester, New York, pp. 77-91, 1997.

[8] **Burns, P. D., Berns, R. S**., Analysis Spectral Image Capture, *Proceedings of IS&T/SID Fourth Color Imaging Conference: Color Science, Systems and Applications*, Scottsdale, AZ, USA, pp 19-22, 1996.

[9] **Commission Internationale de l'Eclairage (CIE),** *Colorimetry*.2nd ed.. Publ. CIE No. 15.2, Bureau Central de la CIE, Austria, 1986.

[10] **Frakas, D. L., Ballou, B. T., Fisher, G. W., Fishman, D., Garini, Y., Niu, W., Wachman, E. S.,** Microscopic and Mesoscopic Spectral Bio-Imaging, *Proceedings of Soc. Photo-Optical Instr. Eng.*, Vol. 2678, pp. 200-206, 1996.

[11] **Friedman, M., Kandel, A.,** *Introduction to Pattern Recognition*, Imperial College Press, London, UK, 1999.

[12] **Gonzalez, R. C., Woods, R. E.,** *Digital Image Processing* Prentice Hall, New York, USA, 2002.

[13] **Haneishi, H., Hasegowa, T., Tsumura, N., Miyake, Y.,** Design of Color Filters for Recording Art Works," *Proceedings of IS&T 50th Annual Conference*, pp.369-372, 1997.

[14] **Hardeberg, J. Y., Schmitt, F., Brettel, H. H., Crettez, J., and Maître, H.,** Multispectral Image Acquisition and Simulation of Illuminant Changes. *Proceedings of MacDonald and Luo,* Vol. 23, pp. 145-164, 1999.

[15] **Hardeberg, J. Y., Schmitt, F., Brettel, H. H., Crettez, J., and Maître, H.,** Spectral Imaging in Multimedia", *Proceedings of CIM'98 Colour Imaging in Multimedia*, University of Derby, UK, pp.75-89, 1998.

[16] **Harding, R. W.,** *Hyperspectral imaging: How much is hype?*, Photonics Spectra, Vol. 31, no. 6, pp. 82-94, 1997.

[17] **Hill, B.,** Spectral Color Technology: A Way Towards High Definition Color Image Scanning and Encoding, *Proceedings of Electronic Image Capture and Publishing EICP'98*, Zurich, Switzerland, pp. 2-13, 1998.

[18] **Hill, B., Vorhagen, F, W**., Spectral Image Pick-Up System, *U.S. Pat.5,319,472,* Germany, 1996.

[19] **Huffman, D.,** A Method for the Construction of Minimum-Redundancy Codes, *Proceedings of the IRE,* Vol. 40, No. 9, pp. 1098-1101, Sep. 1952.

[20] **Kaarna, A., Zemcik, P., Kalviainen, H., Parkkinen., J.** Compression of Multispectral Remote Sensing Images Using Clustering and Spectral Reduction, *Proceedings of the IEEE Transactions on Geosciences and Remote Sensing* . Vol.38, No. 2, pp. 1073-1082, March 2000.

[21] **Keusen, T., Praefcke, W**., Spectral Color System With An Encoding Format Compatible To The Conventional Tristimulus Model, *Proceedings of Third IS&T/SID Color Imaging Conference: Color Science. Systems and Applications*, pp.112-114, 1995.

[22] **König, F, Praefcke W**., A Spectral Scanner, Chapter 7 of *Color Imaging: Vision and Technology*, edited by L. W. *Proceedings of MacDonald and Luo MacDonald and Luo*, pp. 129-143, 1999.

[23] **König, F., Praefcke, W**., A Spectral Scanner, *Proceedings of CIM'98 Color Imaging in Multimedia*, University of Derby, UK, pp. 63-73, 1998.

[24] **König, F., Praefcke, W.,** Practice of Spectral Image acquisition, *Proceedings of Electronic Imaging: Processing, Printing and Publishing in Color, SPIE,* Zurich, Switzerland, Vol. 3409, pp.34-41, 1998.

[25] **König, F., Herzog, P. G.,** On the limitation of spectral imaging, *Proceedings of IS&T's 1999 PICS Conference*, Savannah, GA, USA, pp. 163-168, 1999.

[26] **Louvre, http://www.louvre.fr/**, 8th Apr. 2004.

[27] **Maitre, H., Schmitt, F., Crettez, J.-P., Wu, Y., Hardeberg, J. Y.,** Spectrophotomeric Image Analysis of Fine Art Paintings, *Proceedings of IS&T and SID's 4th Color Imaging Conference: Color Science, Systems and Applications*, Scottsdale, Arizona, pp. 50-53, Nov. 1996.

[28] **Malacara, D**., Color Vision and Colorimetry Theory and Applications, *A publication of SPIE-The International Society for Optical Engineering Bellingham*, Washington USA, London, 1970.

[29] **Manabe, Y., Inokuchi, S.,** Recognition of Material Types and Analysis of Interreflection Using Spectral Image, *Proceedings of the 8th Congress of the International Colour Association, AIC Color 97*, Kyoto, Japan, Vol. 2, pp. 507-510, 1997.

[30] **Manabe, Y., Sato, K., Inkuchi, S**., An object Recognition Through Continuous Spectral Images, *Proceedings of the 12th Internation Conference on Pattern Recognition*, Vol. 1, pp. 858-860, 1994.

[31] **Marimont, D. H., Wandell, B. A.,** Linear Models of Surface and Illuminant Spectra, *Journal of the Optical Society of America*, A, Vol. 9, No.11, pp.1905-1913, 1990.

[32] **Martinez, K., Cupitt, J., and Saunders, D.,** High Resolution Colorimetric Imaging of Paintings, *Proceedings of SPIE*, Vol. 1901, pp. 25–36, Feb. 1993.

[33] **Monnet, G.,** 3D Spectroscopy with Large Telescopes: Past, Present and Prospects, *Proceedings of Tridimensional Optical Spectroscopic Methods in Astronomy,* Vol. 71, pp. 12-17, 1995.

[34] **Museum of Modern Art**, http://www.moma.org/, 8th Apr. 2004.

[35] **National Gallery of Art, Washington**, http://www.nga.gov/, 8th Apr. 2004.

[36] **National Gallery, London**, http://www.nationalgallery.org.uk/**,** 8th Apr. 2004.

[37] **National Museum of Japanese History**, http://www.rekihaku.ac.jp/, 8th Apr. 2004.

[38] **Oja, E.** Subspace Methods of Pattern Recognition, *Proceedings of Pattern Recognition and Image Processing Series*, John Wiley & Sons, Vol. 6, 1983.

[39] **Parkkinen, J., Hallikainen, J., Jaaskelainen, T.,** Characteristic spectra of Munsell colors, *Journal of the Optical Society of America*, Vol. 6, No. 2, pp. 318-322, Feb. 1989.

[40] **Parkkinen, J., Oja, E.,** On Subspace Clustering, *Proceedings of Seventh International Conference of Pattern Recognition*, Montreal, Canada, July 30 – August 2, 1984.

[41] **Peercy, M. S.,** Linear Color Representation for Full Spectral Rendering, *Proceedings of Computer Graphics*, Vol. 27, pp. 191-198, 1993.

[42] **Praefcke, W., Simon, S. F**., Ierarcnische Farbkorrekte Codierung von Reflexionsspektren, *Proceedings of Aachener Symposium .Signaltheorie: Bild und Sprachsignal. Aachen*, Germany, pp. 131-134, March 1997.

[43] **Praefcke, W., Simon, S. F.** Coding of Reflectance Spectra Using Cross-Illumination Color Prediction, *Proceedings of IS&T/SPIE's Symposium on Electronic Imaging*, San Jose, California, USA, Vol. 3018, pp. 130-137, Feb. 1997.

[44] **Praefcke, W.,** Analysis-Synthesis Transforms Versus Orthogonal Transforms for Coding Reflectance Spectra, *Proceedings of Color Imaging Conference*, Scottsdale, AZ. USA, pp. 177-181, Nov. 1997

[45] **Rosselet, A., Graff, W., Wild, U. P., Keller, C. U., Gschwind, R.,** Persistent Spectral Hole Burning Used for Spectrally High-Resolved Imaging of the Sun, *Proceedings of SPIE*, Vol. 2480, pp. 205-212, 1995.

[46] **Saguizbaeva, O.,** *Analysis of Spectral Images of Skin for Medical Application*, M.Sc. thesis, University of Joensuu. Department of Computer Science, 2004.

[47] **Saunders, D., Cupitt, J.,** Image Processing at the National Gallery: The VASARI Project, National Gallery Technical Bulletin, Vol. 14, pp. 72–85, 1993.

[48] **Saunders, D., Hamber, A.,** From Pigments to Pixels: Measurement and Display of the Color Gamut of Paintings, *Proceedings of the SPIE*, Vol. 1250, pp. 90–102, Feb. 1990.

[49] **Saunders, D.,** 10 Years of Art Imaging Research, *Proceedings of the IEEE,* Vol. 90, No. 1 (with J. Cupitt , K. Martinez, and R. Pillay), 2002.

[50] **Savaresi, S. M., Boley, D. L.,** On the performance of bisecting K-means and PDDP, , *Proceedings of First Siam International Conference on Data Mining*, Chicago, USA, April 2001.

[51] **State Hermitage Museum, Saint-Petersburg**, http://www.hermitagemuseum.org/, 8th Apr. 2004.

[52] **Swain, P. H., Davis, S. M**., Remote Sensing: The Quantitative Approach, New York: McGraw-Hill, 1978.

[53] **TIFF™ Revision 6.0 Final,** Aldus Corporation, TIFF Structure- June 3, 1992.

[54] **Viggiano, J. A. S.,** The Comparison of Radiance Ratio Spectra: Assessing a Model's Goodness of Fit, *Proceedings of IS&T's Annual Conference*, Rochester, NY, USA, pp. 225-225, 1990.

[55] **Wang, J., Dodds, Z., Miranker, W**., Principal Component Analysis for Place Recognition, *The Journal of Neural, Parallel, and Scientific Computations*, 1996.

7 APPENDIX

7.1 Colorchecker

7.1.1 Compression results of PCA application with k-means clustering

Table 7.1. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in Spectral space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,12157	0,06978	0,04005	0,03198	0,02457	0,02086
2	0,10670	0,06430	0,03375	0,02457	0,01951	0,01597
3	0,08265	0,05017	0,03138	0,02299	0,01812	0,01469
4	0,07349	0,04896	0,02787	0,01839	0,01524	0,01208
5	0,06810	0,04201	0,02406	0,01759	0,01394	0,01140
6	0,06106	0,03898	0,02266	0,01632	0,01289	0,01046
7	0,05426	0,03154	0,01871	0,01347	0,01087	0,00882
8	0,05685	0,03586	0,02069	0,01568	0,01214	0,00983
9	0,04959	0,02985	0,01930	0,01343	0,01061	0,00868
10	0,04375	0,02370	0,01622	0,01228	0,00978	0,00812
11	0,03876	0,02108	0,01492	0,01153	0,00908	0,00750
12	0,02818	0,01739	0,01363	0,01061	0,00886	0,00741
13	0,03587	0,02244	0,01375	0,01056	0,00867	0,00729
14	0,03148	0,01958	0,01369	0,01042	0,00843	0,00703
15	0,02897	0,01688	0,01318	0,01016	0,00836	0,00693
16	0,03143	0,01747	0,01294	0,01033	0,00814	0,00697
17	0,02961	0,01718	0,01314	0,01012	0,00810	0,00680
18	0,02502	0,01607	0,01271	0,00994	0,00787	0,00679
19	0,02554	0,01618	0,01245	0,00985	0,00805	0,00684
20	0,02290	0,01560	0,01192	0,00928	0,00780	0,00672
21	0,02770	0,01774	0,01350	0,00993	0,00803	0,00683
22	0,02391	0,01670	0,01205	0,00949	0,00787	0,00671
23	0,01975	0,01505	0,01158	0,00913	0,00760	0,00661
24	0,01849	0,01417	0,01097	0,00871	0,00751	0,00644
25	0,02205	0,01459	0,01150	0,00902	0,00758	0,00646

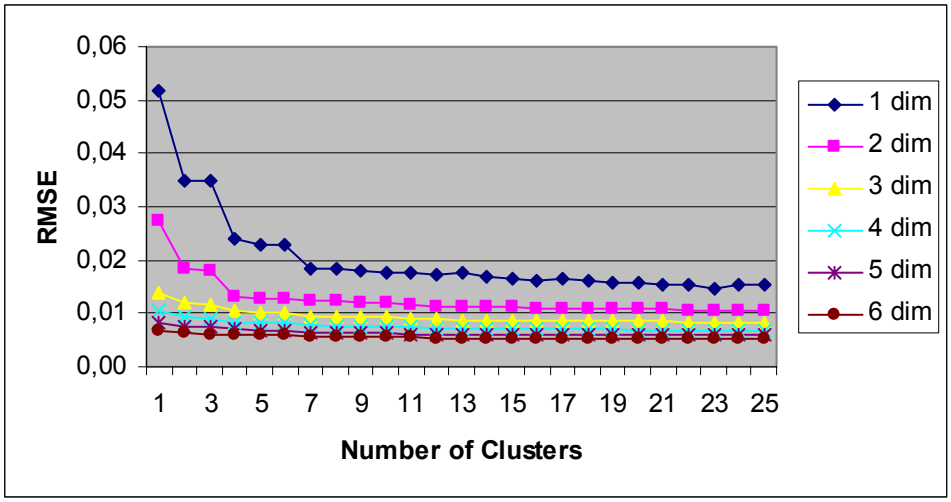

Figure 7.1. RMSE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.2. Color Difference ΔE for 25 clusters and 6 Dimension of the transformation. Image was clustered in Spectral space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	23,75	18,57	6,95	4,87	1,49	1,43
2	20,40	18,09	5,96	2,03	1,39	0,82
3	17,74	12,37	5,87	2,06	1,29	0,72
4	14,10	11,35	4,57	1,91	0,89	0,65
5	11,13	7,79	4,03	1,49	1,17	0,88
6	11,12	6,58	3,43	1,34	1,22	0,86
7	12,49	7,06	1,92	1,36	0,66	0,51
8	9,78	4,18	3,10	1,41	1,07	0,85
9	8,04	4,78	2,93	1,21	0,92	0,83
10	7,13	4,33	2,43	1,21	0,94	0,62
11	7,01	3,16	1,79	1,16	0,83	0,73
12	3,97	2,17	1,49	1,24	0,85	0,72
13	6,24	3,10	1,54	1,24	0,80	0,70
14	4,35	2,40	1,43	1,16	0,77	0,66
15	4,04	2,06	1,40	1,15	0,70	0,63
16	4,25	2,25	1,50	1,19	0,79	0,72
17	4,41	1,96	1,34	1,05	0,77	0,62
18	3,03	2,02	1,39	1,07	0,82	0,74
19	3,42	1,60	1,29	0,98	0,74	0,60
20	3,32	1,57	1,32	0,97	0,78	0,66
21	3,69	1,95	1,46	0,96	0,78	0,63
22	2,90	1,86	1,25	0,99	0,82	0,67
23	2,11	1,41	1,24	0,95	0,77	0,59
24	2,00	1,56	1,23	0,99	0,81	0,68
25	2,29	1,59	1,18	0,93	0,73	0,60

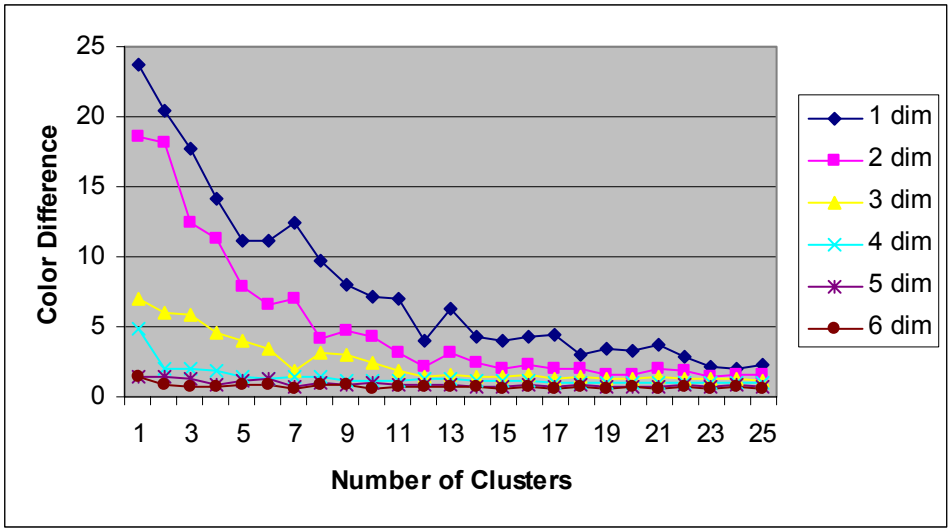

Figure 7.2. Color Difference ΔE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.3. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,12157	0,06978	0,04005	0,03198	0,02457	0,02086
2	0,09446	0,05525	0,03283	0,02486	0,02074	0,01651
3	0,08363	0,05120	0,02779	0,02099	0,01710	0,01358
4	0,06069	0,03002	0,02233	0,01630	0,01285	0,01021
5	0,05652	0,02731	0,01844	0,01382	0,01037	0,00813
6	0,04252	0,02555	0,01537	0,01133	0,00876	0,00742
7	0,04403	0,02507	0,01693	0,01174	0,00922	0,00741
8	0,03226	0,02123	0,01454	0,01137	0,00903	0,00717
9	0,03653	0,02120	0,01293	0,01005	0,00800	0,00705
10	0,03086	0,02007	0,01270	0,00983	0,00795	0,00696
11	0,02506	0,01754	0,01222	0,00982	0,00784	0,00674
12	0,02344	0,01592	0,01152	0,00899	0,00746	0,00665
13	0,02814	0,01805	0,01160	0,00911	0,00751	0,00666
14	0,02970	0,01949	0,01171	0,00904	0,00742	0,00659
15	0,02174	0,01520	0,01098	0,00851	0,00717	0,00641
16	0,02284	0,01473	0,01112	0,00879	0,00732	0,00650
17	0,02110	0,01451	0,01054	0,00825	0,00708	0,00630
18	0,02078	0,01453	0,01071	0,00813	0,00693	0,00625
19	0,02027	0,01418	0,01072	0,00822	0,00692	0,00621
20	0,02028	0,01372	0,01024	0,00787	0,00683	0,00609
21	0,01925	0,01340	0,01013	0,00787	0,00679	0,00610
22	0,01967	0,01370	0,01020	0,00784	0,00675	0,00600
23	0,01945	0,01344	0,01001	0,00778	0,00673	0,00603
24	0,01923	0,01311	0,00990	0,00767	0,00663	0,00593
25	0,01907	0,01320	0,00999	0,00775	0,00669	0,00596

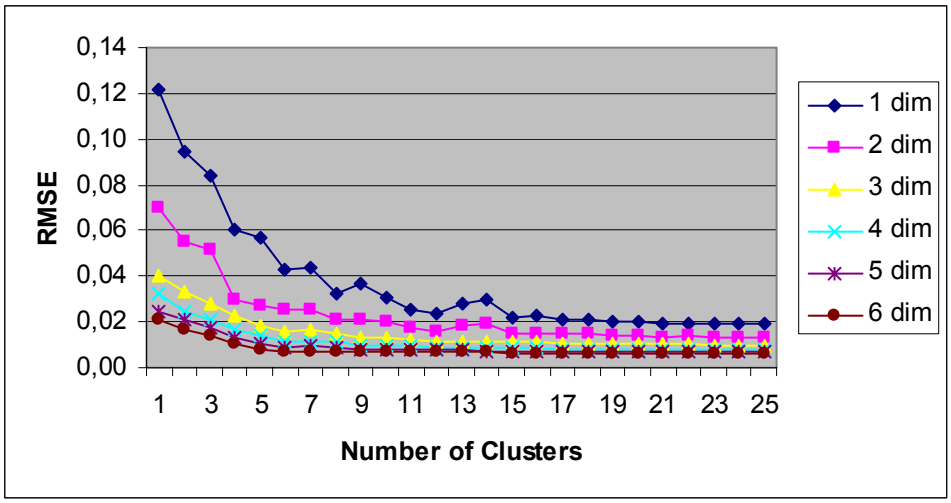

Figure 7.3. RMSE dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.4. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	23,75	18,57	6,95	4,87	1,49	1,43
2	16,65	13,62	6,80	2,71	1,42	0,87
3	16,54	13,60	4,94	2,53	2,88	0,89
4	10,71	6,47	3,03	2,07	0,94	0,53
5	9,53	5,45	2,54	1,42	1,03	0,94
6	6,39	3,79	2,20	1,19	1,09	0,89
7	6,38	3,22	2,22	1,11	1,01	0,90
8	4,18	2,77	1,84	1,42	0,98	0,82
9	5,32	2,78	1,43	1,26	1,18	0,84
10	3,17	1,93	1,37	1,25	1,20	0,83
11	2,81	1,88	1,37	1,12	0,97	0,81
12	2,14	1,43	1,11	1,03	0,97	0,78
13	2,75	1,87	1,15	1,04	0,96	0,72
14	3,54	2,62	1,06	0,99	0,91	0,69
15	1,88	1,32	0,93	0,91	0,85	0,69
16	2,61	1,44	1,19	1,01	0,93	0,72
17	1,70	1,19	0,90	0,87	0,83	0,72
18	1,72	1,20	0,86	0,80	0,75	0,65
19	1,58	1,23	0,87	0,83	0,74	0,63
20	1,49	1,13	0,82	0,79	0,74	0,64
21	1,43	1,15	0,84	0,79	0,73	0,61
22	1,55	1,06	0,74	0,71	0,67	0,60
23	1,51	1,13	0,82	0,76	0,72	0,65
24	1,35	1,03	0,76	0,71	0,66	0,57
25	1,42	1,05	0,76	0,72	0,67	0,60

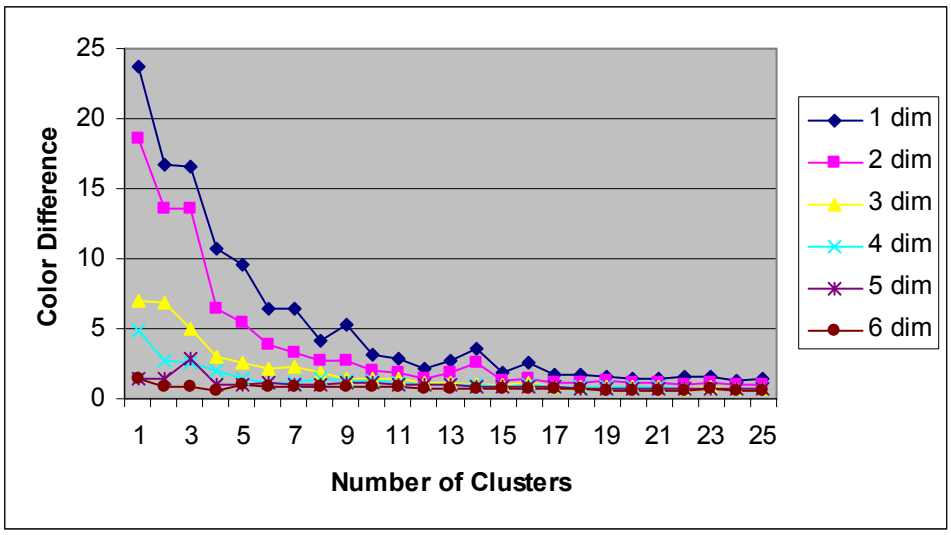

Figure 7.4. Color Difference ΔE dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.5. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,12157	0,06978	0,04005	0,03198	0,02457	0,02086
2	0,11509	0,05494	0,03331	0,02401	0,01954	0,01546
3	0,08095	0,03781	0,02743	0,02011	0,01598	0,01328
4	0,06060	0,03159	0,02230	0,01684	0,01296	0,01029
5	0,06347	0,03548	0,02307	0,01594	0,01186	0,00938
6	0,04900	0,02943	0,01691	0,01254	0,00962	0,00786
7	0,04824	0,02542	0,01584	0,01176	0,00937	0,00749
8	0,04127	0,02066	0,01448	0,01104	0,00906	0,00724
9	0,04259	0,02287	0,01356	0,01054	0,00818	0,00704
10	0,03930	0,01877	0,01287	0,00973	0,00777	0,00683
11	0,03832	0,01820	0,01362	0,01091	0,00882	0,00711
12	0,04198	0,01636	0,01262	0,00961	0,00774	0,00671
13	0,03410	0,01657	0,01226	0,00948	0,00764	0,00676
14	0,03677	0,01771	0,01249	0,00975	0,00773	0,00651
15	0,02883	0,01604	0,01138	0,00901	0,00740	0,00647
16	0,02766	0,01824	0,01251	0,00962	0,00790	0,00664
17	0,02391	0,01503	0,01127	0,00877	0,00730	0,00636
18	0,02302	0,01487	0,01101	0,00870	0,00722	0,00628
19	0,02627	0,01578	0,01108	0,00868	0,00713	0,00625
20	0,02300	0,01390	0,01044	0,00843	0,00707	0,00622
21	0,02173	0,01417	0,01038	0,00830	0,00698	0,00611
22	0,02183	0,01415	0,01073	0,00839	0,00702	0,00620
23	0,02186	0,01309	0,00989	0,00816	0,00698	0,00611
24	0,01801	0,01312	0,00996	0,00806	0,00693	0,00603
25	0,01892	0,01328	0,01015	0,00815	0,00689	0,00606

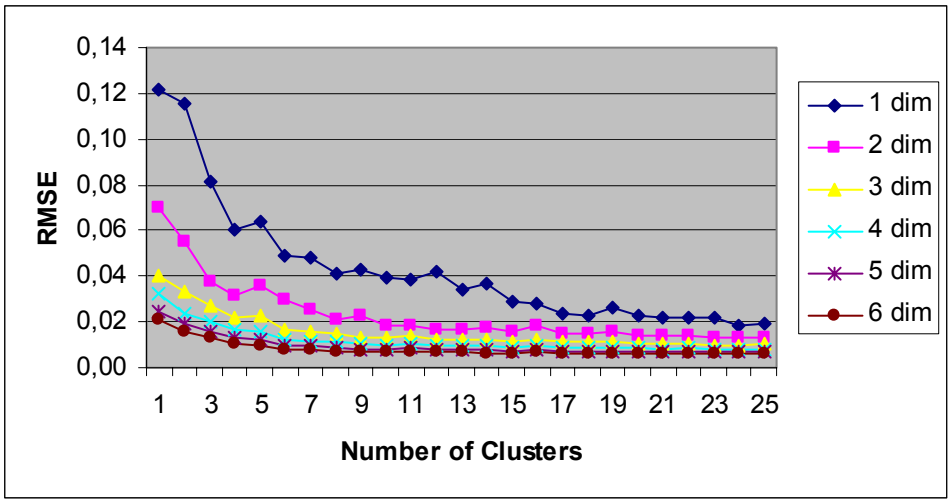

Figure 7.5. RMSE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.

Table 7.6 Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	23,75	18,57	6,95	4,87	1,49	1,43
2	21,74	14,90	7,17	2,59	2,14	0,78
3	15,19	8,02	5,50	1,74	1,55	1,00
4	10,38	6,43	5,05	2,42	1,09	0,78
5	9,66	7,60	4,58	1,86	1,20	0,85
6	8,35	5,06	2,45	1,34	1,02	0,70
7	7,96	4,33	2,55	1,42	0,97	0,62
8	5,63	2,97	2,45	1,51	0,98	0,56
9	6,45	3,15	1,57	1,16	0,99	0,74
10	4,95	2,41	1,30	1,03	0,87	0,68
11	6,10	2,82	2,55	1,40	1,00	0,57
12	5,60	1,65	1,35	1,08	0,83	0,66
13	3,66	2,01	1,63	1,10	0,96	0,72
14	5,38	2,20	1,28	1,10	0,72	0,59
15	3,61	1,86	1,38	1,18	0,78	0,64
16	3,80	2,43	1,20	1,12	0,77	0,68
17	2,74	1,55	1,15	0,95	0,78	0,65
18	2,48	1,58	1,16	1,00	0,79	0,61
19	3,86	2,06	1,07	0,88	0,76	0,59
20	2,31	1,34	1,08	0,94	0,76	0,60
21	2,34	1,48	1,12	0,94	0,76	0,58
22	2,35	1,32	1,01	0,87	0,70	0,59
23	2,06	1,22	1,06	0,96	0,73	0,56
24	1,58	1,16	1,00	0,88	0,74	0,57
25	1,73	1,21	1,02	0,88	0,70	0,58

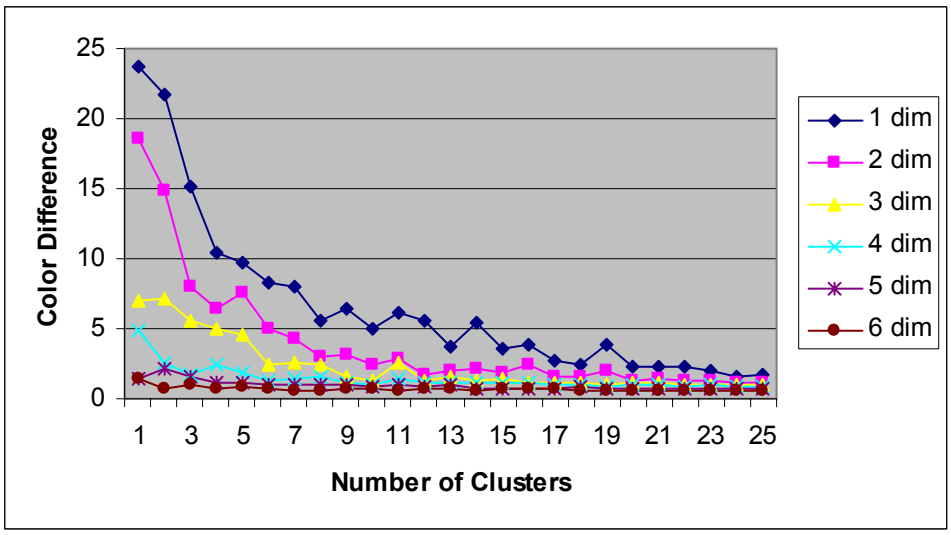

Figure 7.6. Color Difference ΔE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.

7.1.2 Compression results of PCA application with new clustering method.

Table 7.7. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in Spectral space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,12157	0,06978	0,04005	0,03198	0,02457	0,02086
2	0,08476	0,05398	0,03491	0,02595	0,01891	0,01581
3	0,07053	0,04561	0,02650	0,02024	0,01567	0,01320
4	0,06892	0,04299	0,02623	0,01876	0,01416	0,01092
5	0,05131	0,03100	0,01987	0,01416	0,01037	0,00783
6	0,04809	0,02870	0,01743	0,01302	0,00987	0,00790
7	0,04078	0,02588	0,01522	0,01114	0,00849	0,00721
8	0,04246	0,02119	0,01527	0,01092	0,00847	0,00696
9	0,03579	0,01922	0,01333	0,01011	0,00799	0,00679
10	0,03278	0,01887	0,01313	0,01007	0,00795	0,00677
11	0,03231	0,01791	0,01266	0,00954	0,00775	0,00660
12	0,02270	0,01453	0,01101	0,00885	0,00749	0,00654
13	0,02581	0,01547	0,01130	0,00924	0,00769	0,00660
14	0,02475	0,01558	0,01108	0,00878	0,00741	0,00652
15	0,02558	0,01462	0,01127	0,00896	0,00746	0,00648
16	0,02217	0,01335	0,01063	0,00865	0,00728	0,00639
17	0,02369	0,01356	0,01055	0,00868	0,00730	0,00636
18	0,02360	0,01609	0,01110	0,00886	0,00739	0,00648
19	0,02286	0,01469	0,01102	0,00864	0,00728	0,00640
20	0,02064	0,01328	0,01044	0,00857	0,00728	0,00642
21	0,02445	0,01479	0,01073	0,00865	0,00727	0,00640
22	0,01752	0,01232	0,00981	0,00810	0,00704	0,00623
23	0,02184	0,01404	0,01010	0,00822	0,00707	0,00627
24	0,02157	0,01447	0,01110	0,00883	0,00732	0,00630
25	0,01698	0,01236	0,00988	0,00804	0,00698	0,00617

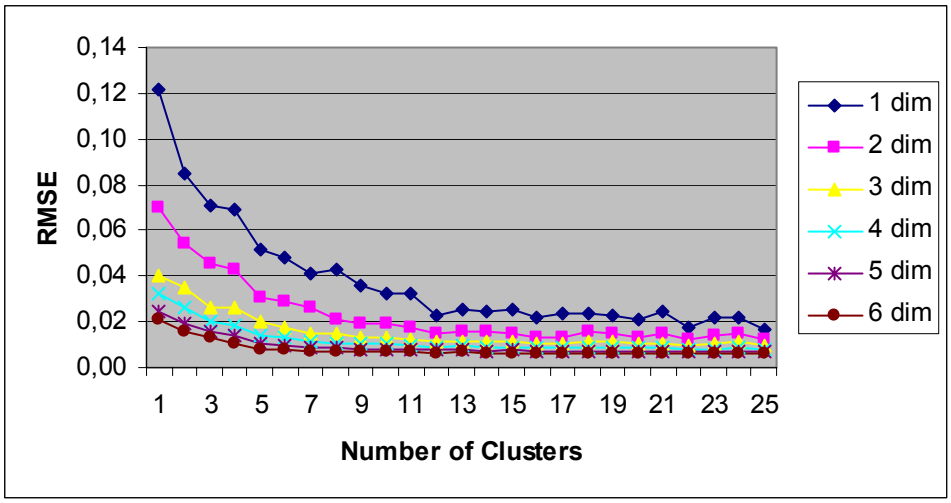

Figure 7.7. RMSE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.8. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in Spectral space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	23,75	18,57	6,95	4,87	1,49	1,43
2	20,63	11,18	4,85	3,43	2,02	0,94
3	16,28	9,77	3,91	1,69	1,39	0,83
4	17,89	7,33	3,89	2,27	0,79	0,52
5	10,51	5,21	3,00	1,55	0,98	0,85
6	9,73	5,22	2,77	1,31	0,86	0,74
7	7,85	4,03	1,75	1,18	1,01	0,84
8	8,24	2,59	1,92	1,16	0,83	0,72
9	6,60	2,45	1,67	1,20	0,91	0,73
10	5,86	2,50	1,69	1,19	0,90	0,73
11	4,82	1,94	1,55	1,22	0,94	0,73
12	2,58	1,39	1,25	1,15	1,00	0,75
13	3,96	1,83	1,39	1,25	0,93	0,72
14	3,68	1,90	1,23	1,14	0,87	0,71
15	3,29	1,39	1,21	1,10	0,91	0,68
16	2,85	1,36	1,25	1,09	0,82	0,65
17	3,69	1,30	1,25	1,03	0,77	0,62
18	3,21	2,10	1,22	1,12	0,93	0,74
19	3,04	1,50	1,06	1,01	0,81	0,65
20	2,86	1,30	1,18	1,06	0,86	0,71
21	4,01	1,95	1,28	1,06	0,76	0,64
22	1,87	1,23	1,12	0,98	0,79	0,62
23	3,20	1,76	1,14	0,97	0,76	0,63
24	3,28	1,67	1,39	0,90	0,74	0,62
25	1,59	1,21	1,10	0,97	0,81	0,62

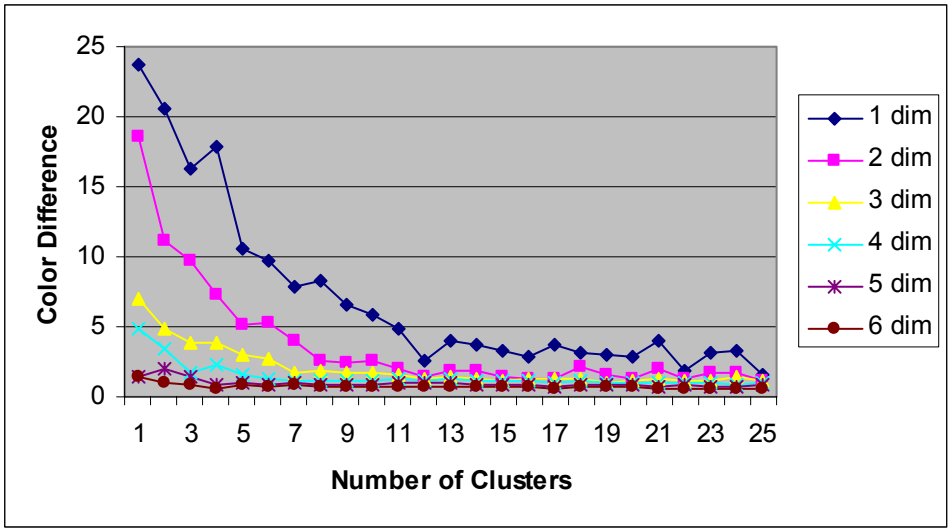

Figure 7.8. Color Difference ΔE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.9. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,12157	0,06978	0,04005	0,03198	0,02457	0,02086
2	0,11317	0,06713	0,03416	0,02323	0,01915	0,01483
3	0,10730	0,04780	0,03097	0,02291	0,01800	0,01393
4	0,09338	0,04267	0,02581	0,01940	0,01425	0,01101
5	0,08529	0,03545	0,02181	0,01586	0,01216	0,01008
6	0,08163	0,03279	0,02162	0,01588	0,01150	0,00900
7	0,07898	0,02894	0,01832	0,01339	0,01037	0,00852
8	0,07835	0,02865	0,01794	0,01320	0,01029	0,00842
9	0,07374	0,02995	0,02033	0,01509	0,01098	0,00856
10	0,07124	0,02347	0,01502	0,01126	0,00907	0,00752
11	0,06369	0,02156	0,01435	0,01100	0,00877	0,00722
12	0,06533	0,02260	0,01437	0,01092	0,00865	0,00724
13	0,06744	0,01951	0,01395	0,01081	0,00843	0,00711
14	0,06116	0,02211	0,01574	0,01141	0,00874	0,00731
15	0,05222	0,01881	0,01336	0,01041	0,00819	0,00700
16	0,06041	0,02020	0,01396	0,01061	0,00842	0,00713
17	0,04158	0,01862	0,01325	0,01033	0,00823	0,00692
18	0,05474	0,02007	0,01347	0,01032	0,00832	0,00697
19	0,04455	0,01992	0,01294	0,01026	0,00807	0,00684
20	0,05511	0,01811	0,01319	0,01025	0,00812	0,00701
21	0,05167	0,01988	0,01304	0,01028	0,00829	0,00697
22	0,04557	0,01793	0,01278	0,00998	0,00794	0,00677
23	0,04082	0,01759	0,01247	0,00999	0,00786	0,00673
24	0,03555	0,01752	0,01227	0,00984	0,00778	0,00668
25	0,03470	0,01780	0,01247	0,00982	0,00783	0,00668

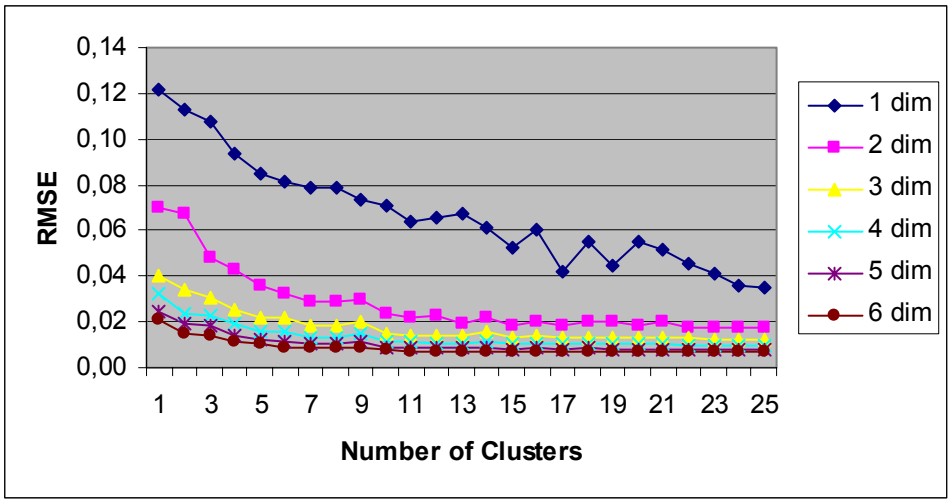

Figure 7.9. RMSE dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.10. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	23,75	18,57	6,95	4,87	1,49	1,43
2	22,52	17,94	6,17	2,53	1,87	1,04
3	22,06	10,85	5,88	3,01	1,37	0,95
4	20,77	8,53	3,96	2,07	1,33	0,88
5	18,64	7,29	2,61	1,58	0,90	0,69
6	17,57	5,29	3,72	1,91	0,83	0,63
7	16,45	5,00	1,92	1,15	0,82	0,61
8	16,37	4,92	1,83	1,11	0,76	0,58
9	14,79	4,82	3,34	1,58	0,76	0,59
10	15,04	4,02	1,25	0,93	0,81	0,53
11	13,59	2,87	1,30	0,89	0,78	0,56
12	12,43	3,78	1,20	0,80	0,70	0,52
13	14,19	2,31	1,25	0,86	0,77	0,53
14	13,13	2,45	1,65	0,77	0,65	0,57
15	10,26	2,13	1,07	0,73	0,65	0,54
16	13,44	2,34	1,16	0,83	0,71	0,57
17	8,38	2,12	1,09	0,76	0,64	0,52
18	10,28	2,32	1,07	0,76	0,59	0,52
19	9,13	3,08	1,10	0,71	0,59	0,49
20	10,81	1,85	1,05	0,81	0,73	0,51
21	10,17	2,31	1,09	0,79	0,62	0,49
22	8,91	1,76	1,05	0,71	0,57	0,50
23	8,39	1,62	1,06	0,73	0,63	0,50
24	7,55	1,63	1,03	0,72	0,58	0,49
25	7,26	1,75	1,04	0,73	0,61	0,49

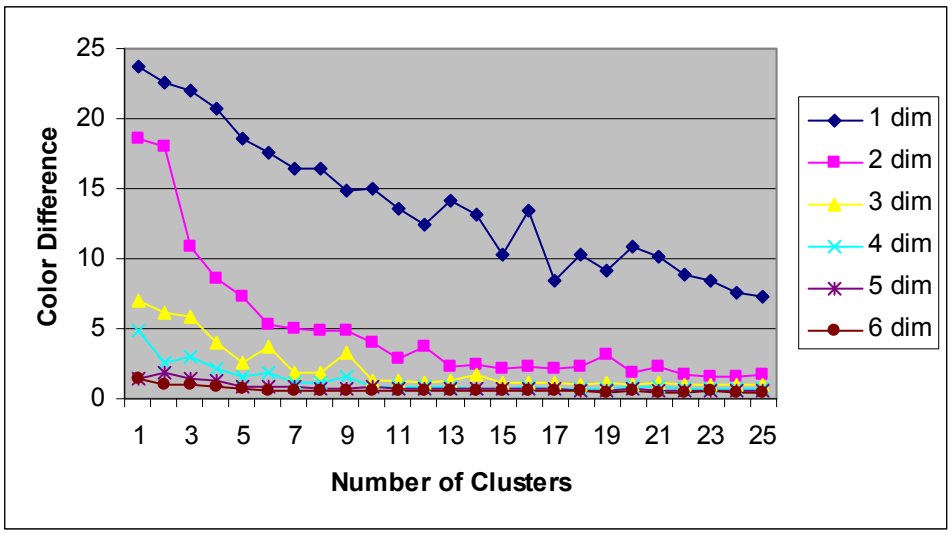

Figure 7.10. Color Difference ΔE dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.11. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,12157	0,06978	0,04005	0,03198	0,02457	0,02086
2	0,09951	0,05346	0,03052	0,02502	0,02084	0,01662
3	0,09102	0,04382	0,02631	0,02060	0,01626	0,01343
4	0,06429	0,03407	0,02310	0,01745	0,01321	0,01073
5	0,05540	0,02778	0,02012	0,01458	0,01084	0,00843
6	0,05168	0,02349	0,01512	0,01179	0,00925	0,00727
7	0,04128	0,02341	0,01580	0,01162	0,00941	0,00776
8	0,03992	0,02155	0,01477	0,01108	0,00867	0,00703
9	0,03441	0,01979	0,01293	0,00980	0,00772	0,00677
10	0,03080	0,01925	0,01308	0,00995	0,00783	0,00674
11	0,03044	0,01744	0,01199	0,00915	0,00743	0,00657
12	0,02929	0,01763	0,01243	0,00945	0,00758	0,00659
13	0,02640	0,01616	0,01156	0,00913	0,00741	0,00656
14	0,02694	0,01497	0,01106	0,00852	0,00724	0,00631
15	0,02470	0,01450	0,01088	0,00844	0,00710	0,00632
16	0,02390	0,01434	0,01059	0,00819	0,00695	0,00620
17	0,02505	0,01461	0,01089	0,00850	0,00708	0,00624
18	0,02467	0,01484	0,01107	0,00849	0,00703	0,00620
19	0,02352	0,01447	0,01092	0,00844	0,00704	0,00626
20	0,02275	0,01364	0,01027	0,00801	0,00681	0,00605
21	0,02237	0,01357	0,01023	0,00798	0,00680	0,00606
22	0,02116	0,01325	0,01002	0,00792	0,00676	0,00603
23	0,02311	0,01358	0,01018	0,00795	0,00674	0,00601
24	0,02110	0,01329	0,01007	0,00786	0,00669	0,00598
25	0,02187	0,01325	0,00993	0,00785	0,00662	0,00593

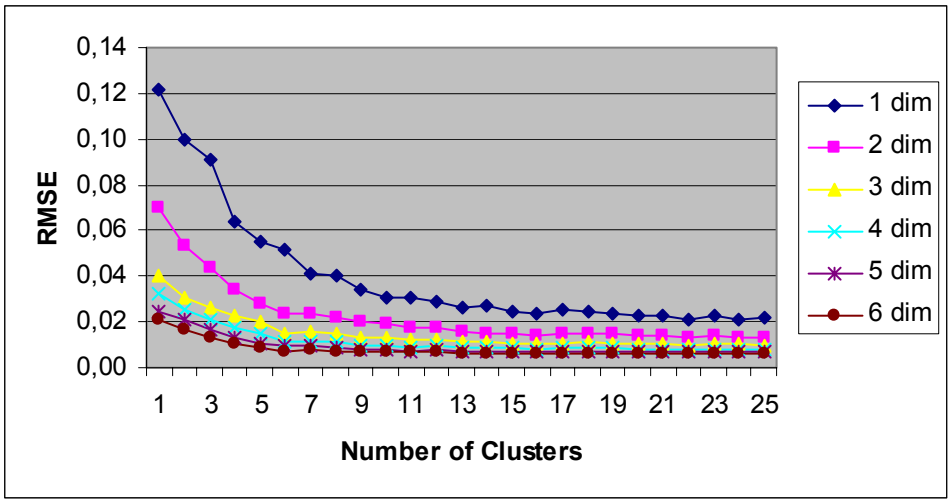

Figure 7.11. RMSE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.

Table 7.12. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	23,75	18,57	6,95	4,87	1,49	1,43
2	18,86	13,30	4,89	2,70	1,78	1,27
3	17,16	10,46	3,97	2,26	2,06	1,04
4	11,10	5,60	3,86	2,22	0,95	0,68
5	10,06	4,96	3,06	1,26	1,09	0,97
6	7,82	2,64	1,88	1,82	0,87	0,83
7	7,45	2,80	1,95	1,59	1,37	1,00
8	6,75	2,60	1,89	1,07	0,92	0,86
9	4,82	2,14	1,31	1,10	0,94	0,75
10	4,07	2,33	1,55	1,07	0,94	0,73
11	3,79	2,03	0,97	0,93	0,84	0,71
12	3,11	2,04	1,28	0,95	0,84	0,71
13	3,06	1,36	1,12	1,02	0,93	0,77
14	3,17	1,10	0,89	0,88	0,76	0,68
15	2,29	1,12	0,91	0,88	0,79	0,69
16	2,31	1,04	0,85	0,83	0,73	0,64
17	2,47	1,12	0,86	0,83	0,74	0,65
18	2,34	1,09	0,86	0,79	0,70	0,62
19	2,13	1,11	0,89	0,86	0,78	0,68
20	1,95	0,98	0,84	0,77	0,68	0,61
21	1,84	0,98	0,81	0,77	0,68	0,62
22	1,63	1,00	0,85	0,78	0,74	0,65
23	2,01	0,91	0,79	0,72	0,63	0,56
24	1,57	0,95	0,78	0,72	0,64	0,56
25	1,83	0,88	0,76	0,65	0,61	0,55

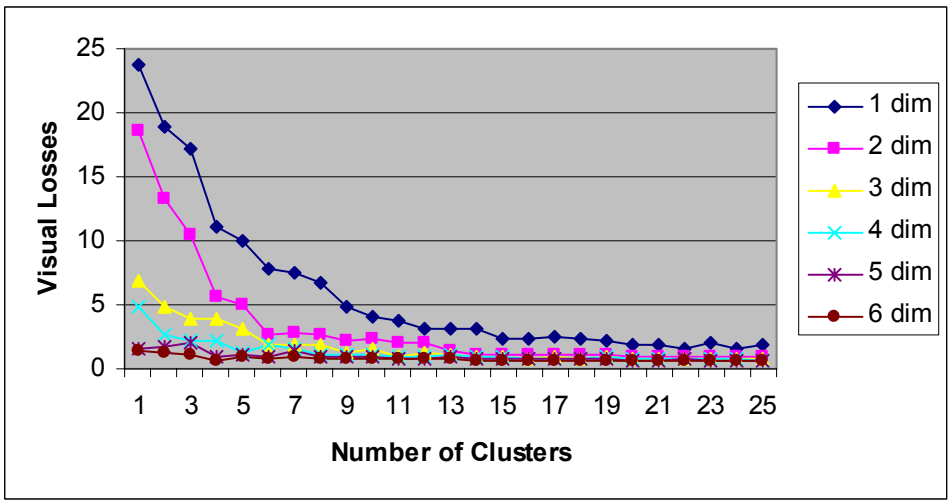

Figure 7.12. Color Difference ΔE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.

7.2 Icon

7.2.1 Compression results of PCA application with k-means clustering

Table 7.13. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in Spectral space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,05187	0,02725	0,01406	0,01059	0,00822	0,00690
2	0,03593	0,01964	0,01266	0,00988	0,00769	0,00631
3	0,02995	0,01875	0,01224	0,00948	0,00747	0,00623
4	0,02704	0,01673	0,01150	0,00883	0,00736	0,00618
5	0,02621	0,01610	0,01125	0,00858	0,00711	0,00609
6	0,02286	0,01484	0,01086	0,00835	0,00691	0,00594
7	0,02239	0,01471	0,01080	0,00833	0,00687	0,00589
8	0,02177	0,01475	0,01058	0,00827	0,00674	0,00588
9	0,02037	0,01361	0,01011	0,00817	0,00668	0,00583
10	0,01916	0,01305	0,00990	0,00805	0,00660	0,00574
11	0,01911	0,01312	0,00998	0,00803	0,00658	0,00569
12	0,01831	0,01277	0,00977	0,00796	0,00655	0,00569
13	0,01855	0,01274	0,00968	0,00776	0,00652	0,00562
14	0,01817	0,01258	0,00961	0,00770	0,00644	0,00560
15	0,01759	0,01232	0,00949	0,00760	0,00635	0,00557
16	0,01740	0,01229	0,00944	0,00755	0,00631	0,00553
17	0,01715	0,01213	0,00934	0,00763	0,00630	0,00551
18	0,01668	0,01212	0,00942	0,00754	0,00632	0,00554
19	0,01654	0,01199	0,00937	0,00750	0,00628	0,00550
20	0,01631	0,01219	0,00937	0,00751	0,00625	0,00552
21	0,01613	0,01185	0,00925	0,00745	0,00623	0,00549
22	0,01600	0,01177	0,00924	0,00746	0,00621	0,00547
23	0,01595	0,01177	0,00918	0,00734	0,00619	0,00545
24	0,01561	0,01164	0,00914	0,00741	0,00620	0,00546
25	0,01538	0,01159	0,00917	0,00740	0,00617	0,00544

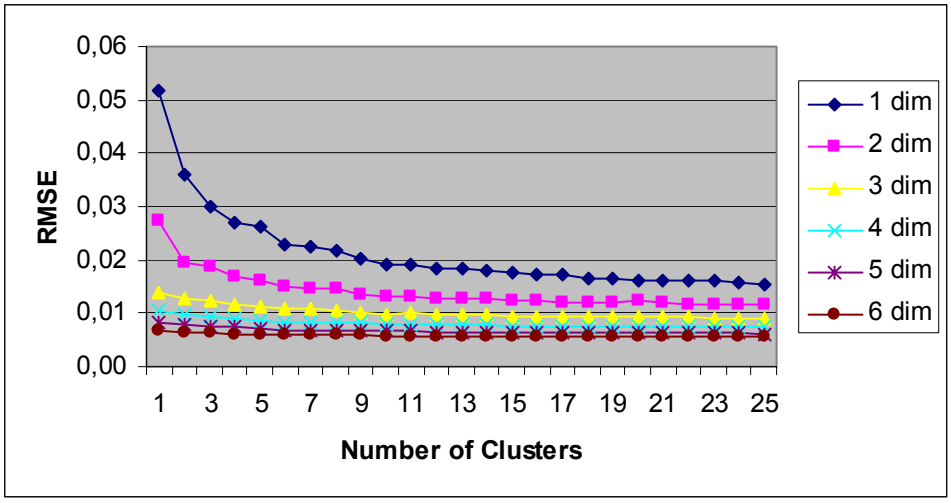

Figure 7.13. RMSE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.14. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in Spectral space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	18,96	11,28	6,98	2,56	1,16	1,05
2	9,53	8,14	5,78	3,54	1,26	1,14
3	8,06	7,00	4,94	2,96	1,75	1,10
4	7,88	5,78	3,90	2,79	2,16	1,15
5	7,26	5,41	3,48	2,66	1,96	1,21
6	6,01	4,62	3,27	2,45	1,88	1,11
7	5,99	4,54	3,28	2,44	1,91	1,09
8	5,70	4,45	3,35	2,35	1,83	1,10
9	4,88	4,01	2,94	2,27	1,83	1,07
10	4,62	3,76	2,79	2,16	1,89	1,07
11	4,59	3,74	2,85	2,17	1,88	1,14
12	4,25	3,49	2,67	2,17	1,73	1,25
13	4,35	3,41	2,64	2,24	1,84	1,18
14	4,16	3,30	2,59	2,14	1,78	1,15
15	4,02	3,12	2,52	2,01	1,79	1,21
16	3,94	2,98	2,47	2,00	1,70	1,18
17	3,91	2,95	2,40	1,97	1,70	1,20
18	3,85	2,94	2,42	1,97	1,74	1,17
19	3,79	2,91	2,41	1,91	1,69	1,13
20	3,70	3,08	2,45	1,94	1,63	1,22
21	3,62	2,90	2,41	1,86	1,70	1,16
22	3,64	2,87	2,42	1,86	1,68	1,14
23	3,54	2,79	2,35	1,89	1,65	1,22
24	3,67	2,87	2,42	1,85	1,65	1,15
25	3,51	2,89	2,35	1,81	1,64	1,20

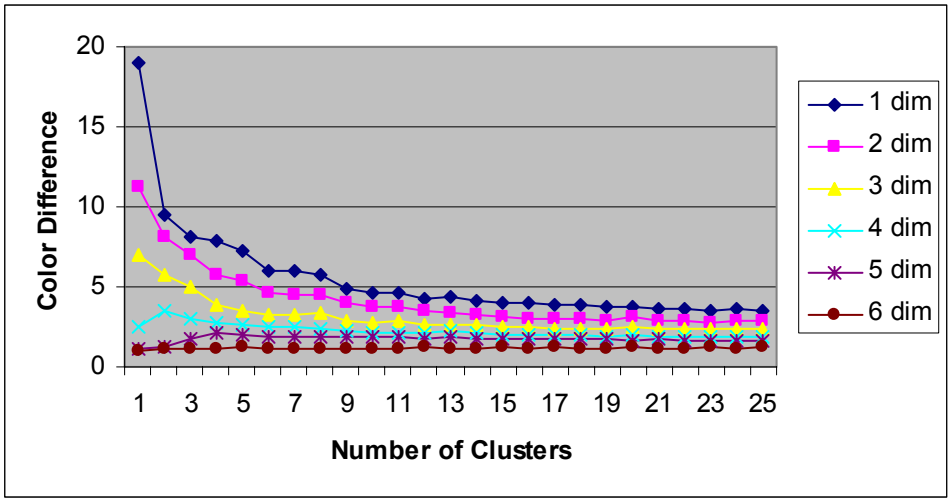

Figure 7.14. Color Difference ΔE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.15. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,05187	0,02725	0,01406	0,01059	0,00822	0,00690
2	0,04448	0,01727	0,01181	0,00945	0,00795	0,00671
3	0,03471	0,01518	0,01161	0,00912	0,00757	0,00626
4	0,03073	0,01348	0,01066	0,00834	0,00700	0,00596
5	0,02792	0,01294	0,01013	0,00817	0,00683	0,00586
6	0,02671	0,01260	0,00985	0,00812	0,00678	0,00579
7	0,02528	0,01239	0,00952	0,00775	0,00659	0,00571
8	0,02484	0,01215	0,00933	0,00769	0,00654	0,00567
9	0,02342	0,01170	0,00917	0,00757	0,00638	0,00560
10	0,02428	0,01173	0,00903	0,00746	0,00632	0,00563
11	0,02381	0,01128	0,00881	0,00731	0,00621	0,00552
12	0,02367	0,01119	0,00878	0,00726	0,00616	0,00547
13	0,02353	0,01117	0,00872	0,00721	0,00613	0,00544
14	0,02327	0,01103	0,00856	0,00716	0,00607	0,00542
15	0,02277	0,01116	0,00866	0,00717	0,00605	0,00537
16	0,02278	0,01114	0,00863	0,00714	0,00603	0,00537
17	0,02106	0,01099	0,00848	0,00709	0,00596	0,00533
18	0,02194	0,01099	0,00845	0,00701	0,00592	0,00531
19	0,02192	0,01095	0,00841	0,00695	0,00590	0,00529
20	0,02184	0,01092	0,00838	0,00690	0,00588	0,00528
21	0,02077	0,01084	0,00834	0,00687	0,00585	0,00526
22	0,02123	0,01083	0,00833	0,00687	0,00586	0,00526
23	0,02081	0,01080	0,00831	0,00682	0,00582	0,00524
24	0,01971	0,01075	0,00829	0,00679	0,00582	0,00522
25	0,02062	0,01076	0,00826	0,00679	0,00580	0,00521

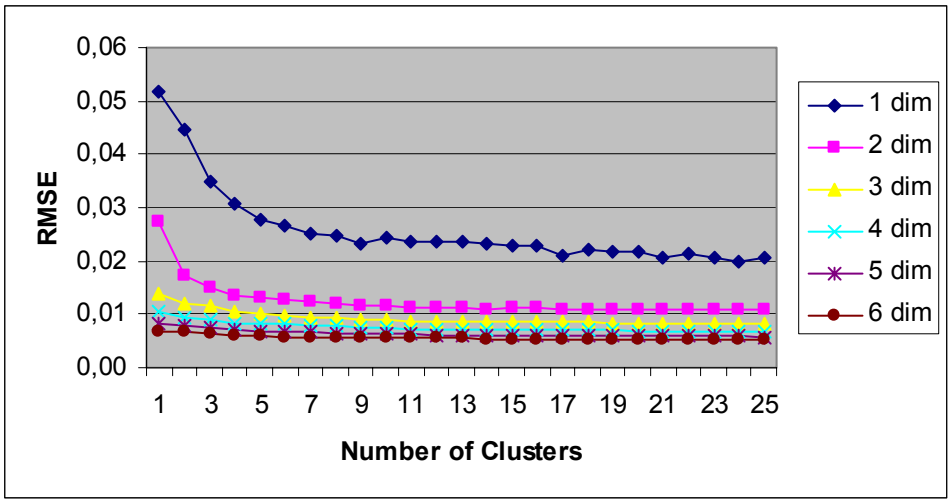

Figure 7.15. RMSE dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.16. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	18,96	11,28	6,98	2,56	1,16	1,05
2	13,25	6,68	3,41	3,31	1,79	1,25
3	8,56	5,72	3,63	3,48	1,82	1,32
4	6,93	4,19	3,01	2,19	1,94	1,47
5	5,76	3,63	2,79	2,15	1,95	1,34
6	5,40	3,42	2,83	2,13	1,87	1,24
7	5,14	3,19	2,55	2,27	1,84	1,25
8	4,91	2,95	2,44	2,26	1,83	1,22
9	4,44	2,50	2,38	2,11	1,93	1,27
10	4,60	2,48	2,19	2,01	1,79	1,31
11	4,52	2,19	2,17	1,96	1,75	1,37
12	4,45	2,11	2,06	1,93	1,71	1,31
13	4,39	2,07	2,07	1,88	1,66	1,29
14	4,00	1,95	1,94	1,80	1,62	1,33
15	4,06	1,98	1,95	1,78	1,59	1,26
16	4,02	1,97	1,93	1,76	1,57	1,26
17	3,59	1,86	1,83	1,66	1,49	1,26
18	3,62	1,85	1,79	1,68	1,49	1,25
19	3,58	1,81	1,76	1,68	1,47	1,24
20	3,54	1,79	1,73	1,66	1,46	1,22
21	3,57	1,76	1,71	1,61	1,44	1,26
22	3,47	1,78	1,72	1,62	1,46	1,22
23	3,35	1,71	1,66	1,57	1,40	1,20
24	3,20	1,68	1,63	1,55	1,38	1,19
25	3,29	1,67	1,60	1,57	1,36	1,17

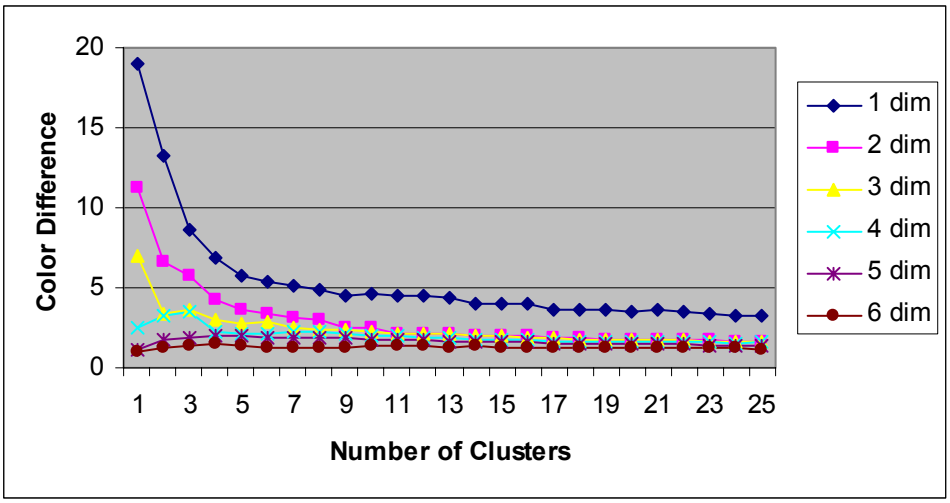

Figure 7.16. Color Difference dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.17. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,05187	0,02725	0,01406	0,01059	0,00822	0,00690
2	0,03483	0,01832	0,01201	0,00939	0,00763	0,00626
3	0,03475	0,01787	0,01169	0,00908	0,00741	0,00618
4	0,02412	0,01325	0,01054	0,00833	0,00698	0,00592
5	0,02276	0,01282	0,01026	0,00819	0,00684	0,00585
6	0,02279	0,01282	0,01011	0,00811	0,00682	0,00583
7	0,01841	0,01249	0,00955	0,00776	0,00654	0,00566
8	0,01834	0,01243	0,00951	0,00761	0,00644	0,00561
9	0,01792	0,01206	0,00931	0,00747	0,00629	0,00553
10	0,01765	0,01195	0,00919	0,00743	0,00626	0,00550
11	0,01758	0,01169	0,00909	0,00735	0,00618	0,00547
12	0,01707	0,01128	0,00891	0,00723	0,00612	0,00543
13	0,01746	0,01143	0,00881	0,00718	0,00604	0,00537
14	0,01677	0,01110	0,00874	0,00714	0,00605	0,00541
15	0,01642	0,01106	0,00867	0,00711	0,00603	0,00539
16	0,01622	0,01103	0,00864	0,00708	0,00602	0,00537
17	0,01637	0,01100	0,00863	0,00704	0,00596	0,00531
18	0,01613	0,01089	0,00855	0,00697	0,00592	0,00529
19	0,01589	0,01082	0,00854	0,00697	0,00591	0,00526
20	0,01569	0,01078	0,00845	0,00691	0,00588	0,00526
21	0,01549	0,01079	0,00849	0,00693	0,00588	0,00526
22	0,01544	0,01068	0,00841	0,00689	0,00587	0,00524
23	0,01445	0,01054	0,00840	0,00690	0,00588	0,00526
24	0,01524	0,01054	0,00835	0,00681	0,00582	0,00520
25	0,01520	0,01055	0,00836	0,00680	0,00581	0,00521

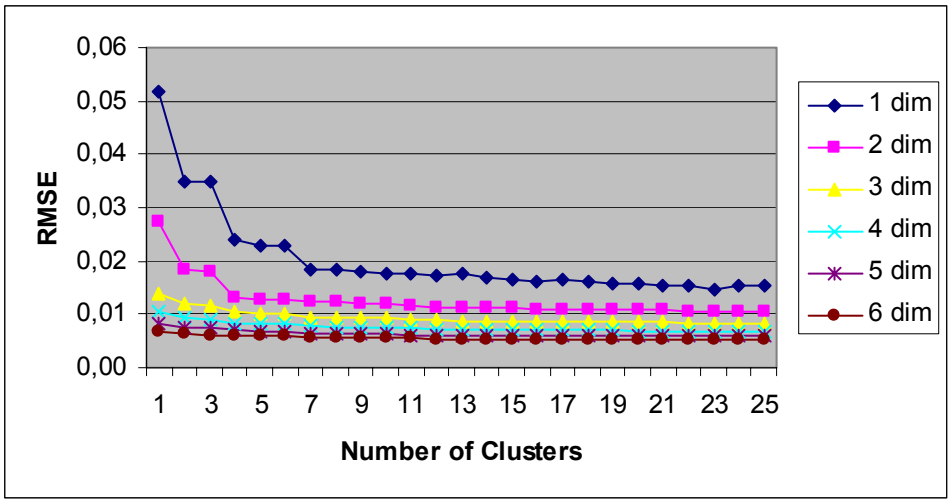

Figure 7.17. RMSE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.

Table 7.18. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	18,96	11,28	6,98	2,56	1,16	1,05
2	9,45	7,52	4,47	3,76	1,41	1,19
3	8,89	6,96	3,92	3,37	1,59	1,22
4	6,02	4,16	3,32	2,18	2,03	1,34
5	5,12	3,46	3,21	1,99	1,89	1,40
6	5,09	3,43	2,82	2,10	1,98	1,36
7	4,05	3,22	2,76	2,16	1,89	1,29
8	3,93	3,19	2,51	2,14	2,01	1,19
9	3,68	2,92	2,53	2,04	1,88	1,22
10	3,62	2,85	2,46	2,01	1,85	1,25
11	3,43	2,66	2,34	1,96	1,76	1,25
12	3,20	2,55	2,24	1,88	1,67	1,27
13	3,27	2,42	2,18	1,86	1,63	1,28
14	3,07	2,40	2,16	1,89	1,65	1,30
15	3,01	2,38	2,13	1,86	1,61	1,30
16	2,93	2,34	2,09	1,81	1,60	1,29
17	2,90	2,26	2,05	1,76	1,53	1,27
18	2,84	2,24	1,99	1,77	1,49	1,27
19	2,77	2,22	1,98	1,70	1,45	1,18
20	2,68	2,20	1,93	1,75	1,48	1,24
21	2,68	2,23	1,96	1,71	1,46	1,26
22	2,62	2,17	1,89	1,75	1,46	1,24
23	2,55	2,20	1,94	1,70	1,45	1,15
24	2,54	2,09	1,83	1,65	1,38	1,17
25	2,48	2,07	1,82	1,62	1,39	1,17

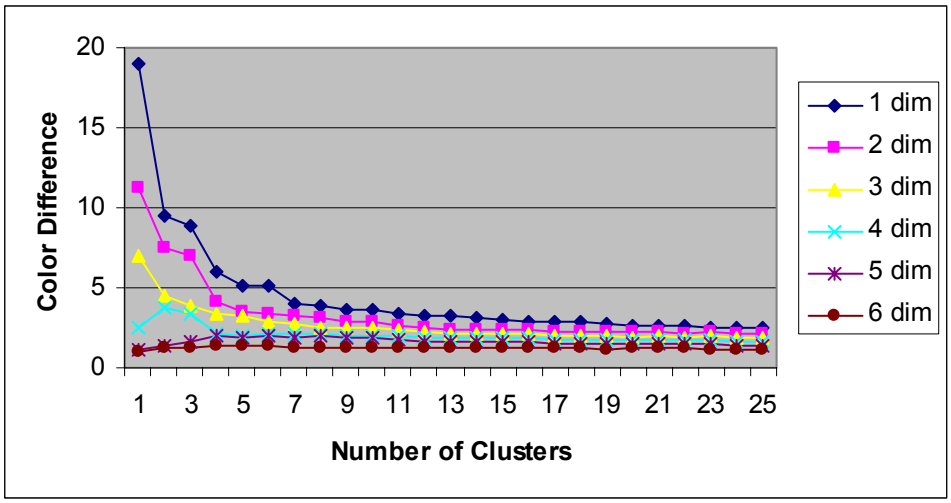

Figure 7.18. Color Difference ΔE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.

7.2.2 Compression results of PCA application with Subspace Clustering method

Table 7.19. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in Spectral space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,05187	0,02725	0,01406	0,01059	0,00822	0,00690
2	0,03312	0,01638	0,01169	0,00928	0,00769	0,00635
3	0,02227	0,01367	0,01082	0,00844	0,00712	0,00603
4	0,01949	0,01336	0,01063	0,00825	0,00694	0,00599
5	0,01754	0,01322	0,01051	0,00821	0,00694	0,00593
6	0,01644	0,01276	0,01042	0,00817	0,00682	0,00589
7	0,01564	0,01159	0,00936	0,00776	0,00644	0,00563
8	0,01478	0,01146	0,00930	0,00769	0,00637	0,00559
9	0,01421	0,01141	0,00919	0,00763	0,00631	0,00555
10	0,01372	0,01108	0,00918	0,00758	0,00631	0,00556
11	0,01373	0,01101	0,00909	0,00753	0,00628	0,00552
12	0,01327	0,01095	0,00901	0,00748	0,00628	0,00551
13	0,01312	0,01081	0,00891	0,00751	0,00629	0,00552
14	0,01286	0,01052	0,00875	0,00738	0,00616	0,00547
15	0,01287	0,01059	0,00873	0,00736	0,00619	0,00548
16	0,01251	0,01048	0,00873	0,00729	0,00613	0,00544
17	0,01240	0,01016	0,00852	0,00725	0,00609	0,00538
18	0,01223	0,01018	0,00853	0,00719	0,00608	0,00541
19	0,01207	0,01005	0,00840	0,00711	0,00602	0,00534
20	0,01197	0,00998	0,00845	0,00715	0,00607	0,00539
21	0,01186	0,00987	0,00838	0,00712	0,00604	0,00533
22	0,01183	0,00987	0,00835	0,00713	0,00605	0,00536
23	0,01173	0,00979	0,00832	0,00711	0,00604	0,00536
24	0,01160	0,00964	0,00820	0,00698	0,00599	0,00532
25	0,01144	0,00958	0,00819	0,00695	0,00596	0,00530

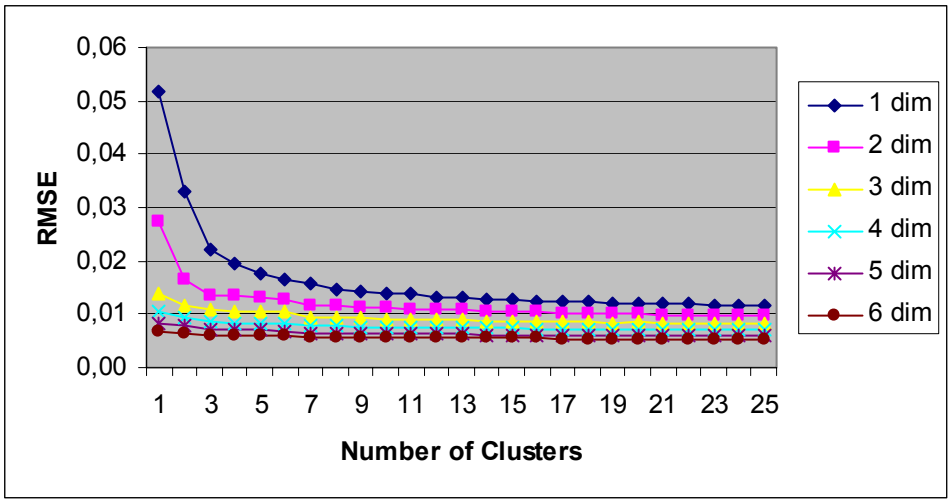

Figure 7.19. RMSE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.20. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in Spectral space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	18,96	11,28	6,98	2,56	1,16	1,05
2	10,62	6,11	3,75	3,45	1,59	1,33
3	7,01	4,13	2,75	2,46	2,22	1,18
4	7,14	3,87	2,77	2,35	2,21	1,20
5	5,34	3,76	2,94	2,36	2,22	1,23
6	5,13	3,71	2,99	2,28	2,16	1,20
7	3,91	2,87	2,60	2,05	1,59	1,33
8	4,01	2,71	2,48	2,03	1,51	1,25
9	3,41	2,77	2,56	1,99	1,48	1,18
10	3,38	2,72	2,36	1,85	1,50	1,21
11	3,57	2,64	2,31	1,88	1,53	1,22
12	3,24	2,63	2,32	1,90	1,53	1,21
13	3,23	2,63	2,43	1,78	1,54	1,20
14	3,19	2,68	2,31	1,79	1,49	1,19
15	3,16	2,64	2,36	1,86	1,58	1,18
16	2,97	2,55	2,22	1,94	1,46	1,17
17	2,96	2,62	2,30	1,74	1,49	1,16
18	2,92	2,61	2,27	1,83	1,44	1,17
19	2,81	2,55	2,25	1,96	1,47	1,17
20	2,83	2,65	2,29	1,98	1,42	1,17
21	2,72	2,53	2,26	1,92	1,48	1,14
22	2,82	2,53	2,19	1,86	1,45	1,17
23	2,80	2,54	2,21	1,90	1,50	1,13
24	2,66	2,54	2,23	1,93	1,42	1,14
25	2,63	2,50	2,26	1,93	1,43	1,14

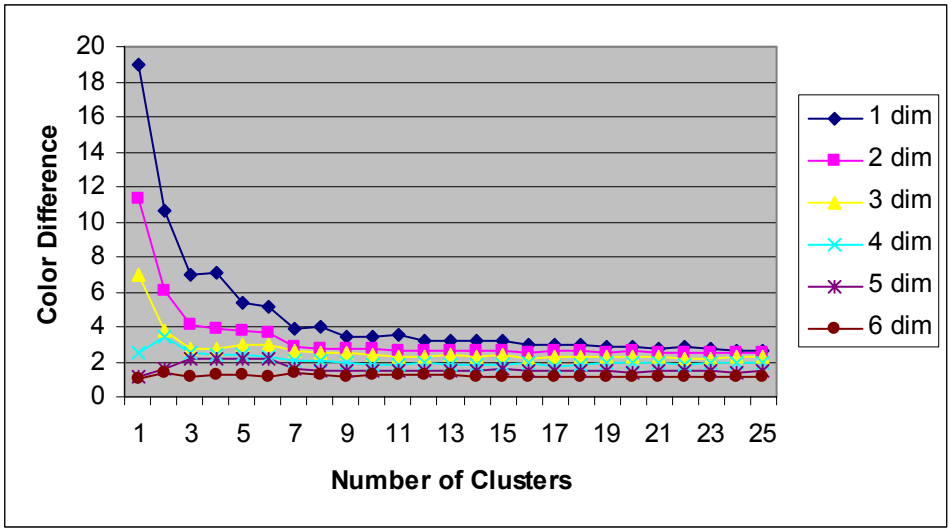

Figure 7.20. Color Difference ΔE dependency from Number of Clusters. Image was clustered in Spectral space.

Table 7.21. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,05187	0,02725	0,01406	0,01059	0,00822	0,00690
2	0,04297	0,02265	0,01162	0,00929	0,00801	0,00669
3	0,04024	0,01959	0,01110	0,00868	0,00750	0,00637
4	0,03842	0,01933	0,01101	0,00863	0,00744	0,00631
5	0,03984	0,01911	0,01079	0,00835	0,00712	0,00607
6	0,03842	0,01820	0,01072	0,00828	0,00699	0,00597
7	0,03702	0,01714	0,01054	0,00808	0,00668	0,00584
8	0,03714	0,01806	0,01062	0,00820	0,00679	0,00585
9	0,03598	0,01797	0,01055	0,00814	0,00681	0,00583
10	0,03499	0,01660	0,01039	0,00797	0,00656	0,00572
11	0,03596	0,01724	0,01041	0,00798	0,00659	0,00570
12	0,03443	0,01692	0,01034	0,00792	0,00653	0,00565
13	0,03420	0,01696	0,01033	0,00791	0,00650	0,00562
14	0,03507	0,01709	0,01035	0,00795	0,00657	0,00566
15	0,03499	0,01727	0,01035	0,00794	0,00657	0,00564
16	0,03412	0,01669	0,01028	0,00789	0,00649	0,00562
17	0,03394	0,01669	0,01028	0,00787	0,00645	0,00559
18	0,03392	0,01665	0,01026	0,00787	0,00644	0,00558
19	0,03380	0,01661	0,01025	0,00787	0,00645	0,00558
20	0,03367	0,01650	0,01024	0,00784	0,00643	0,00557
21	0,03379	0,01663	0,01025	0,00786	0,00645	0,00557
22	0,03343	0,01638	0,01020	0,00781	0,00641	0,00555
23	0,03369	0,01642	0,01022	0,00783	0,00642	0,00555
24	0,03367	0,01636	0,01022	0,00783	0,00641	0,00555
25	0,03328	0,01627	0,01019	0,00780	0,00639	0,00554

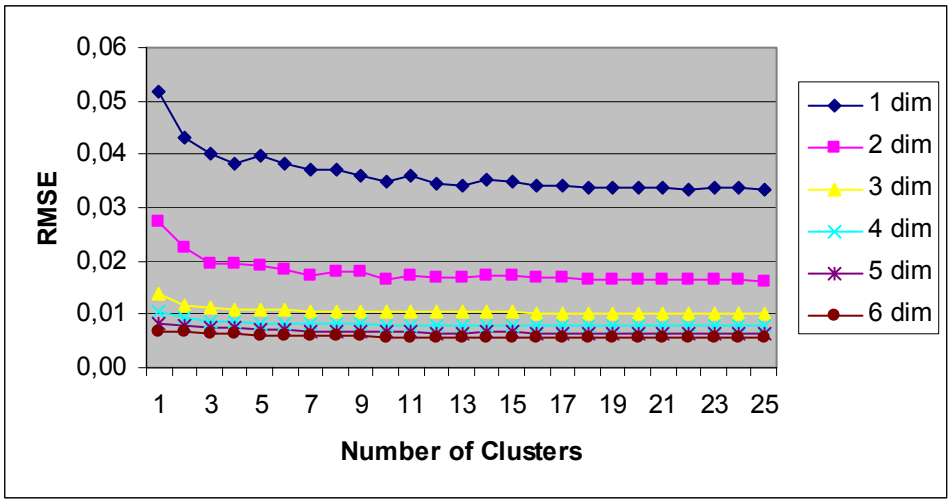

Figure 7.21. RMSE dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.22. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in a^*b^* space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	18,96	11,28	6,98	2,56	1,16	1,05
2	13,56	8,99	2,96	3,09	1,71	1,06
3	12,77	6,52	2,19	1,99	1,48	1,36
4	11,79	6,25	2,17	2,00	1,43	1,31
5	12,39	6,28	1,83	1,57	1,32	1,14
6	12,22	5,64	1,75	1,48	1,24	1,08
7	11,43	4,87	1,67	1,39	1,23	1,07
8	12,01	5,70	1,76	1,40	1,21	1,01
9	11,17	5,33	1,67	1,37	1,15	0,95
10	10,75	4,34	1,61	1,32	1,17	0,95
11	11,38	4,93	1,66	1,22	1,12	0,89
12	10,82	4,68	1,62	1,25	1,09	0,85
13	10,81	4,69	1,62	1,23	1,07	0,82
14	10,94	4,76	1,62	1,18	1,07	0,84
15	10,98	5,08	1,60	1,17	1,06	0,81
16	10,67	4,63	1,60	1,18	1,04	0,80
17	10,66	4,43	1,61	1,17	1,02	0,77
18	10,67	4,41	1,59	1,16	1,01	0,76
19	10,59	4,54	1,59	1,16	1,01	0,75
20	10,55	4,38	1,58	1,13	0,99	0,74
21	10,63	4,51	1,59	1,14	0,99	0,74
22	10,49	4,24	1,59	1,11	0,98	0,73
23	10,60	4,32	1,59	1,12	0,97	0,72
24	10,59	4,29	1,60	1,12	0,98	0,72
25	10,40	4,22	1,55	1,10	0,97	0,71

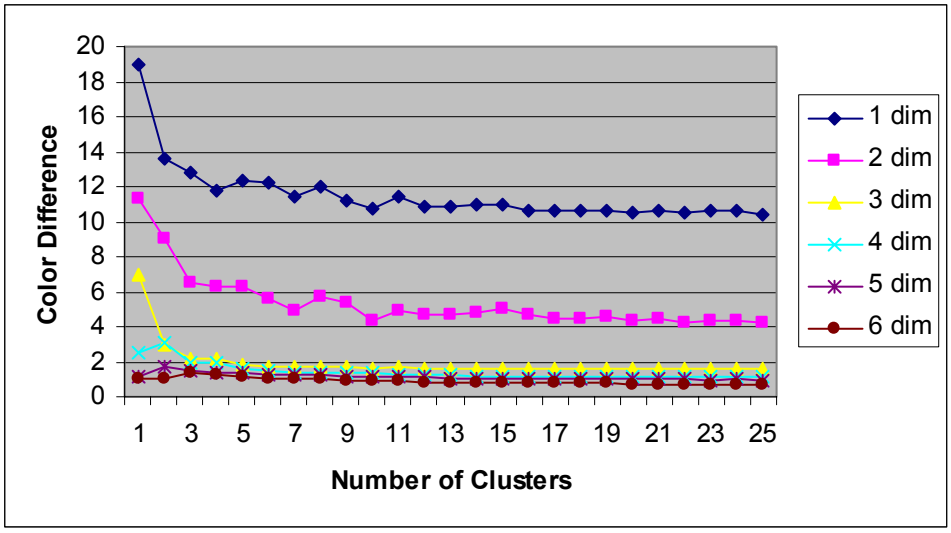

Figure 7.22. Color Difference ΔE dependency from Number of Clusters. Image was clustered in a^*b^* space.

Table 7.23. RMSE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	0,05187	0,02725	0,01406	0,01059	0,00822	0,00690
2	0,03546	0,01788	0,01144	0,00888	0,00747	0,00625
3	0,03015	0,01408	0,01066	0,00856	0,00715	0,00612
4	0,02758	0,01305	0,00993	0,00839	0,00698	0,00592
5	0,02399	0,01291	0,00967	0,00812	0,00676	0,00583
6	0,02639	0,01220	0,00944	0,00782	0,00669	0,00577
7	0,02150	0,01221	0,00933	0,00766	0,00650	0,00567
8	0,02104	0,01196	0,00915	0,00755	0,00645	0,00562
9	0,02112	0,01174	0,00895	0,00738	0,00635	0,00559
10	0,02028	0,01156	0,00875	0,00722	0,00623	0,00552
11	0,02027	0,01147	0,00869	0,00714	0,00615	0,00548
12	0,01985	0,01140	0,00862	0,00705	0,00606	0,00538
13	0,01929	0,01132	0,00857	0,00701	0,00602	0,00537
14	0,01905	0,01124	0,00853	0,00695	0,00597	0,00535
15	0,01872	0,01112	0,00847	0,00694	0,00595	0,00534
16	0,01860	0,01112	0,00843	0,00692	0,00593	0,00530
17	0,01844	0,01105	0,00840	0,00688	0,00590	0,00528
18	0,01825	0,01094	0,00834	0,00681	0,00586	0,00526
19	0,01899	0,01097	0,00833	0,00682	0,00584	0,00526
20	0,01833	0,01098	0,00830	0,00679	0,00583	0,00524
21	0,01846	0,01086	0,00825	0,00672	0,00582	0,00522
22	0,01813	0,01091	0,00831	0,00679	0,00584	0,00525
23	0,01823	0,01076	0,00819	0,00669	0,00575	0,00517
24	0,01805	0,01076	0,00817	0,00667	0,00574	0,00515
25	0,01831	0,01075	0,00820	0,00670	0,00575	0,00518

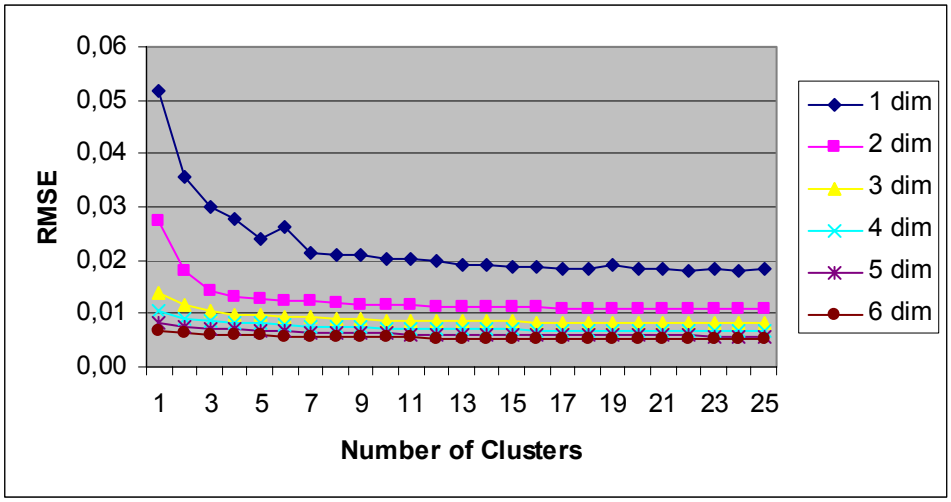

Figure 7.23. RMSE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.

Table 7.24. Color Difference ΔE for 25 clusters and 6 Dimensions of the transformation. Image was clustered in $L^*a^*b^*$ space.

	1 dim	2 dim	3 dim	4 dim	5 dim	6 dim
1	18,96	11,28	6,98	2,56	1,16	1,05
2	9,33	7,24	3,19	3,08	2,64	1,21
3	6,50	4,46	3,09	2,56	2,44	1,45
4	5,63	3,72	2,94	2,53	1,97	1,43
5	4,65	3,46	2,60	2,30	1,99	1,40
6	5,32	2,93	2,58	2,34	1,90	1,31
7	4,02	2,87	2,38	2,14	1,95	1,32
8	3,87	2,64	2,28	2,17	1,91	1,28
9	3,80	2,40	2,11	2,03	1,88	1,35
10	3,60	2,29	2,06	1,93	1,78	1,31
11	3,55	2,17	1,96	1,83	1,69	1,18
12	3,38	2,08	1,87	1,74	1,60	1,23
13	3,23	2,03	1,82	1,72	1,60	1,26
14	3,18	1,96	1,77	1,66	1,55	1,24
15	3,07	1,88	1,75	1,63	1,52	1,22
16	3,01	1,85	1,69	1,59	1,50	1,19
17	2,96	1,82	1,66	1,55	1,47	1,17
18	2,96	1,77	1,64	1,54	1,41	1,19
19	3,03	1,74	1,58	1,48	1,37	1,13
20	2,86	1,72	1,55	1,45	1,37	1,13
21	2,90	1,67	1,53	1,44	1,34	1,13
22	2,82	1,65	1,53	1,46	1,36	1,15
23	2,85	1,58	1,47	1,39	1,29	1,09
24	2,81	1,56	1,44	1,37	1,26	1,06
25	2,80	1,51	1,42	1,34	1,24	1,07

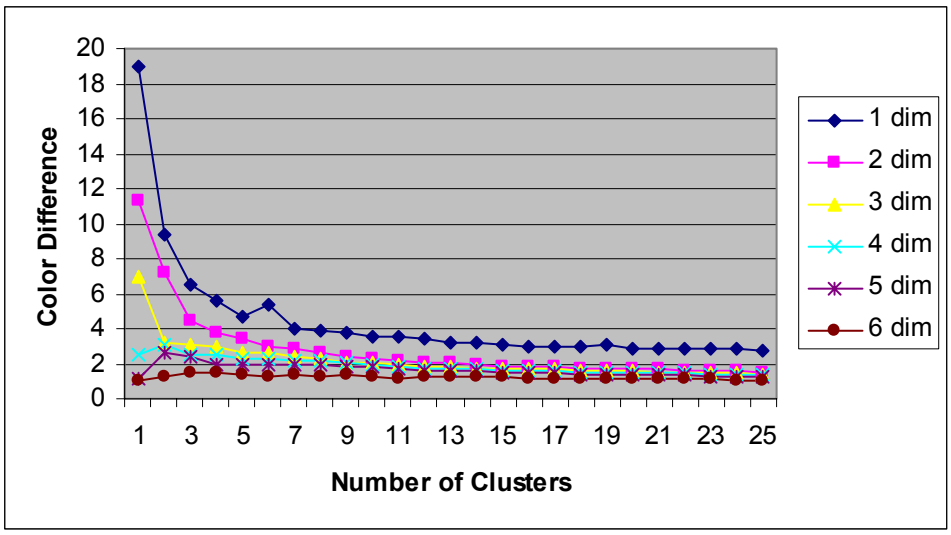

Figure 7.24. Color Difference ΔE dependency from Number of Clusters. Image was clustered in $L^*a^*b^*$ space.