Abstract—This paper proposes a novel approach in image compression based on Local Binary Pattern (LBP). LBP has already been used as a simple texture descriptor, labeling the image pixels by looking at the points surrounding a central point (usually on a 3x3 neighborhood) and examining whether these neighbors' color values are greater or less than the central point and accordingly assigning a binary value to the corresponding bit. The description of image's local pattern results in an eight-bit binary description, but in order to restore the image from such a LBP description, the value of each central pixel is also needed. These two pieces of information, i.e. the LBP description and the actual original value for each local neighborhood central pixel, are stored in a newly proposed Local Binary Compressed format, denoted .LBC, from which the image can be reconstructed by employing statistical methods, i.e. generating smaller or larger sets of random numbers to fill in the missing information within each local neighborhood, based on the LBP descriptor. Two statistical distributions were tested and, apart from the compression performance, a Structural Similarity Index Metric was used to evaluate the results.

I. INTRODUCTION

Nowadays, an amazing amount of data are generated every moment, e.g. in a single minute Google receives two million search requests and Facebook users share more than 684,000 pieces of content, while more than 144.8 billion e-mails are sent during a day [1]. The global Internet population has grown 6.59 percent from 2010 to 2011 and now represents 2.1 billion people, all face the issue of more and more storage resources required every day, e.g. Instagram photographers share 3,600 new photos and 3,125 photos are being uploaded on Flickr [2]. Compared to 2010, the amount of data grew more than 50 times, at the end of 2012 there were 550 billion images, and it is estimated to be approximately 40 zeta bytes (ZB) by 2020 [3]. Considering an average of 20MB/photo and a compression rate of 0.15, the predicted quantity of archived information is about 550 billion * 20MB / 0.15 = 73 exabytes [4].

Human kind needs more and more space to store the big data generated every day and arriving from multiple sources at an alarming velocity, volume and variety. Every digital process and social media produces data, while systems, sensors, and mobile devices transmit them. Unfortunately, traditional data management and analytical tools seem unable to reliably handle and store it all.

The emerging concept of Big Data requires new methods and new approaches for data archiving and storing, as well as real time data analyzing, so we foresee new tools in image processing and analysis to be released in the very next years. Leading organizations are developing new roles, focusing on key challenges and creating new business models to gain the most from Big Data [4].

This paper is structured in the following way: in section II theoretical foundations for local binary pattern (LBP) and structural similarity metric (SSIM) are presented; section III discusses the image compression techniques, while the subsequent sections focus on the original work, results, and conclusions regarding the proposed approach for image compression.

II. THEORETICAL FOUNDATIONS

Images have been and will always be of paramount importance for human communication. The visual system, which helps identify and classify objects, is considered the most important means in analyzing the environment.

The need to classify and analyze images forced scientists and mathematicians to create new methods and tools to archive and store large image resources. Although the Internet has been the first option for searching and retrieving information and the clouds already are an important data storing solution, even the Internet companies face the issue of more and more storage resources required every day.

A. Local Binary Pattern (LBP)

The concept of Local Binary Pattern (LBP) was introduced by Ojala [5] as a fine scale texture descriptor, used to summarize the local structure of images. In fact, it is the particular case of the Texture Spectrum model previously proposed [5, 6].

LBP labels the image pixels and creates a binary number used for classification in computer vision [7, 8]. This method “looks” at each pixel and compares it with its neighbours' colour value. LBP is tolerant to monotonic illumination changes, an important advantage being its computational simplicity, therefore making possible real-time analyzes in challenging settings.

As Fig. 1 shows, the LBP description is created by dividing the image into small 3x3 pixel matrices. The colour value of each central pixel is then compared with
its eight neighbours to "see" whether these neighbors' color values are greater or less than the central point and a binary value is accordingly assigned to the corresponding bit.

Figure 1. The LBP descriptor (adapted from [6]). The 90 value is stored as the LBP description, and the 88 value as the central point information. Together, they are contained in the .LBC format.

The algorithm is applied on a 3x3 neighbourhood, so for each central point there are eight neighbours, leading to an eight-bit value and a subsequent distinct label (\(2^8 = 256\) different labels), used as a local texture descriptor. This simple operator, using only integer arithmetics, proved to be computationally efficient and effective for real-time challenging problems. Moreover, it is invariant to brightness changes and robust to variations of illumination conditions [7-11].

B. Structural Similarity Index Metric (SSIM)

During compression, transmission, storing, and processing, images are prone to visual quality progressive degrading. However, having them assessed by a person is time-consuming and ineffective. The alternative is to employ automated procedures and objective methods able to evaluate the image quality and predict the human perception. Moreover, such techniques are valuable tools for dynamic monitoring and quality adjustments.

According to the availability of the undistorted original image, the quality evaluation metrics can be classified into two categories:

- full-reference → reference image exists;
- no-reference → there is no reference image.

Most applications have to find the balance between computational efficiency and reconstruction accuracy, so developing methods to compare them and assess images' quality has focused important efforts [12-16].

Among common methods employed for such evaluation were Peak Signal-to-Noise Ratio (PSNR) and Root-Mean-Square Deviation (RMSD) [12], but their popularity has declined, for they do not consider the perceptual differences between the compared pictures.

Objective methods for assessing the quality of an image have attempted to quantify the visibility of errors between a distorted image and a reference image by implementing mechanisms borrowed from the human visual system. At present, an accepted instrument for quality evaluation is Structural Similarity Index Metric (SSIM) [13], a metric based on the degradation of the structural information. There were earlier versions of this approach, which had promising results following some simple tests and the SSIM algorithm was generalized for a broader set of validation results. Surface luminance is the product of the observed object illumination and the reflection, but the objects' structure is independent of the landscape lighting.

The purpose of the algorithm is to separate the influence of illumination. The structural information in the image is defined by attributes of average luminance and contrast [12-16].

III. RELATED WORK. IMAGE COMPRESSION TECHNIQUES

Image compression is aimed at reducing the data quantity, without degrading the image quality beyond an acceptable threshold. This can be achieved by removing the redundancy present in the image. In computer science and in information theory, data compression is the process of encoding information using fewer bits than the encoded representation, with the advantage of reducing the consumption of significant resources such as disk space or transmission bandwidth.

Implementing the compression of an image requires storing the image in a bit flow/stream as compact as possible and decoding the image as accurately as possible. The needed elements are an encoder and a decoder. The encoder receives the image and converts it into a series of binary data which are then transmitted or stored. The decoder re-creates the image as accurately as possible. The flow compression is described in Fig. 2.

The important properties of a compression algorithm are the compression ratio and the reconstruction quality. The compression ratio is the report of bits numbers needed to represent the data before and after compression.

The techniques for image compression have to settle a compromise between performance and quality, so they are either lossless (i.e. completely reversible) or "lossy" (i.e. irreversible) methods. The former does not cause any loss in image quality and is used in cases where image accuracy is very important such as technical drawings. The latter handles things differently, aiming at a higher compression rate, so the reconstructed image differs from the original, and the effectiveness is assessed with methods and algorithms meant to estimate the differences between the two versions. We further detail some known image compression techniques.

Run-Length Encoding (RLE) was a simple technique based on the usual repetition of colours in an image. e.g. when reading rows of pixels (i.e. the image from the top left corner, downwards), quite often there are rows of the same colour. When encountering more than three consecutive pixels of the same colour in a row, store the number of pixels and the colour. Using this technique, no information is lost. RLE was used in .PCX format which no longer exists.

The Standard Joint Photographic Experts Group (JPEG) specified the algorithm used for compression and decompression (codec). Depending on the concrete image properties, this format involves some loss of some visual information.

Figure 2. The process of image compression
Based on the fact that the human visual system is insensitive to small colour changes and with a good compression rate, the JPG format is unsuitable for high-contrast images such as screenshots or computer art. It is based on the fact that the eye is not sensitive to small colour changes. Moreover, compression can vary when the image is stored, this formatting algorithm re-compressing each time the image is saved. These repeated compressions could result in a dramatic quality loss. Therefore, the actual work should always be carried out on the uncompressed images, before saving them into the final target format.

Graphics Interchange Format (GIF) is based on limiting the colours used in the image. Usually up to 256 colours are used to achieve the colour palette. i.e. a table assigning colours to a range of 256 numbers (from 0 to 255). The image pixels are then stored using an 8-bit number that represents the colour position in the table. This format is suitable for cartoons and computer art, and supports image transparency.

Portable Network Graphics (PNG) uses the lossless data compression. PNG is an open-source format that was created to improve the GIF. GIF was not an open source format and developers had to pay for the license to use it. This was a motivating factor in the assimilation of the PNG file.

Scalable Vector Graphics (SVG) is a language for describing 2D graphics and graphical applications in XML. It is primarily used for vector graphics on the World Wide Web.

Overall, in the context of the emerging big data approaches and pervasive mobile technology, image compression is a dynamic field, now focussing on challenges like obtaining accurate models of images, optimal models' representations, and their subsequently fast processing [17-22].

IV. THE PROPOSED METHOD

The aim of the compression approach herewith presented was to reach high compression rates without dramatic loss of the original information. The application is currently processing greyscale images, and therefore the image chosen by the user is converted into a greyscale image (Fig. 3).

From an original image, two smaller images are created: one made of LBP codes (Fig. 4), the other containing the values for the neighbourhood's centre pixels (Fig. 5). Pixel greyscale image is divided into 3x3 matrices. If 3 does not divide into the image width or/and height, the border matrices are filled in with the value 0. For every 3x3 matrix, the LBP code is calculated for every central gray value in each neighbourhood, and resulting in a eight-bit binary code (i.e. 1 if the pixel value is equal or greater than the central pixel and 0 otherwise) [9-11].

For further storing and/or transmission, the image-couple is used.

The initial greyscale image is reconstructed by employing statistical methods and making use of this format consisting of two components bearing complementary information: (I) the central information for the 3x3 neighbourhood, i.e. the initial value of the central pixel; (ii) the dispersion of the local contrast, i.e. the LBP.

Two statistical approaches for reconstruction have been tested:

- Reconstruction based on uniform distribution with a flat percentage (Fig. 6) – based on the LBP, the user-specified percentage is subtracted/added to restore the 3x3 matrices;
- Reconstruction based on Gaussian distribution (Fig. 7) – based on the LBP, the values to be restored are individually generated following a normal distribution and according to the pre-specified limits.

Figure 3. Changing a colour image into grayscale was the first step in the testing process

Figure 4. The image bearing the LBP information

Figure 5. The image bearing the 3x3 neighbourhood central information

Figure 6. Reconstructed image based on 2% uniform distribution

Figure 7. The image bearing the LBP information
Applying this technique leads to a new representation for compressed images – we called it the Local Binary Compressed (.LBC) format.

To assess the quality of the reconstruction and compare the performance of the two statistical models, the algorithm of Structural Similarity Index Metric (SSIM) was used.

V. EXPERIMENTAL RESULTS

The proposed method was implemented and tested in a Java application. In the following, we illustrate the results on images with different characteristics: (a) a pencil sketch, (b) a landscape photograph, (c) a portrait photograph, (d) an ultrasound medical image (Fig. 8).

For testing, the reconstruction was performed on each image using multiple percentage values for dispersion [15, 16]. Table I summarizes the results.

However, results proved that the lower the percentage, the better the accuracy and similarity between images. The exception was the pencil sketch, for it contains many fine lines. Although the SSIM values might look small, overall, the best results were obtained around the value of about 5% for the dispersion. At least at this stage, the uniform approach seems to reach better results but, as expected, the SSIM value heavily depends on the image content/characteristics.

An important advantage of the approach is the compression rate, especially when considered in the context of good SSIM values. Table II shows a comparison between the .JPG and .LBC formats, with promising compression indices, i.e. over 3.5.

VI. CONCLUSIONS

Technology advances at an increasing pace, still surpassed by the commercial market and the social networking platforms, both increasingly demanding new solutions for picture handling, storing, and processing [1-4]. Medical applications relying on huge image databases, the personal health records requiring real-time operation, the patient-centred healthcare already a reality, all put forward new challenges for image handling [23, 24].

A possible and promising solution is image compression and reliable restoration. This paper proposes a new technique for compressing an image based on the Local Binary Pattern operator.

<table>
<thead>
<tr>
<th>Image (Fig.8)</th>
<th>Distribution of generated values</th>
<th>Limits of dispersion around the central pixel (17%, 9%, 6%, 2%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) uniform</td>
<td>0.511 0.6702 0.7315 0.6723</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.5992 0.6984 0.6949 0.6143</td>
<td></td>
</tr>
<tr>
<td>(b) uniform</td>
<td>0.6364 0.7353 0.7776 0.8094</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.4603 0.6396 0.7255 0.8024</td>
<td></td>
</tr>
<tr>
<td>(c) uniform</td>
<td>0.7261 0.8778 0.9177 0.9194</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.3601 0.6489 0.7981 0.9249</td>
<td></td>
</tr>
<tr>
<td>(d) uniform</td>
<td>0.6547 0.8452 0.9188 0.9635</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.3682 0.6654 0.8143 0.9519</td>
<td></td>
</tr>
</tbody>
</table>

For testing, the reconstruction was performed on each image using multiple percentage values for dispersion [15, 16]. Table I summarizes the results.

For testing, the reconstruction was performed on each image using multiple percentage values for dispersion [15, 16]. Table I summarizes the results.
The main advantage of this new approach consists of the good compression rates, i.e. between 3.49 and 4.18.

A drawback might be the information loss, still an issue at this stage of testing, although not always visible at the human examination.

However, the SSIM used to evaluate the restoring quality showed encouraging results, i.e. values up to 0.96, so motivating further efforts to be put in developing new techniques for image reconstruction.

REFERENCES