This is a preprint of an article accepted for publication in *Pattern Analysis and Applications* 24(4):1475-1487, 2021. The final authenticated version is available online at: https://doi.org/10.1007/s10044-021-00999-5.

Last compiled on October 24, 2021.

Testing the randomness of shares in color visual cryptography

Leszek J. Chmielewski · **Mariusz Nieniewski** · **Arkadiusz Orłowski**

Abstract The concept of black and white visual cryptography with two truly random shares, previously applied to color images, was improved by mixing the contents of the segments of each coding image, and by randomly changing a specified number of black pixels into color ones. This was done in such a way that the changes of the contents of the decoded image were as small as possible. These modifications made the numbers of color pixels in the shares close to balanced, which potentially made it possible for the shares to be truly random. The true randomness was understood as that the data pass the suitably designed randomness tests. The randomness of the shares was tested with the NIST randomness tests. Part of the tests passed successfully, while some failed. The target of coding a color image in truly random shares was approached, but not yet reached. In visual cryptography the decoding with the unarmed human eye is of primary importance, but besides this, simple numerical processing of the decoded image makes it possible to greatly improve the quality of the reconstructed image, so that it becomes close to that of the dithered original image.

Keywords Visual cryptography · color images · random shares · random coding · randomness tests · NIST

Leszek J. Chmielewski
Warsaw University of Life Sciences – SGGW
Institute of Information Technology
Nowoursynowska 159, 02-775 Warsaw, Poland
E-mail: leszek_chmielewski@sggw.edu.pl – corresponding author

Mariusz Nieniewski
University of Lodz
Faculty of Mathematics and Informatics
Banacha 22, 90-238 Łódź, Poland
E-mail: mariusz.nieniewski@wmii.uni.lodz.pl

Arkadiusz Orłowski
Warsaw University of Life Sciences – SGGW
Institute of Information Technology
Nowoursynowska 159, 02-775 Warsaw, Poland
E-mail: arkadiusz_orlowski@sggw.edu.pl

1 Introduction

The methods of visual cryptography are used to transfer the visual information in a safe way, so that no technical device is needed to decode the secret message. The secret message, or simply the secret, is an image. This secret is coded in a pair of images, from which none contains any information on the secret. When these images are overlaid on one another, the secret image becomes visible to an unarmed human eye.

In the classical visual cryptography introduced by Naor and Shamir [20,21] the secretness of the coded image results from the lack of correlation between the secret and the images in which the secret is coded. This lack of correlation is sufficient for the secret to be impossible to reconstruct from any single coding image. However, the coding images have the characteristics which reveal the mere fact that some information has been hidden.

These fundamental works have been extended to gray-level and also color images in a number of ways (see [5,18,25] and the literature therein). To name some recent works let us refer to [6,8] where images of good quality were effectively obtained with the CMY and RGB color models. The use of numerical post-processing to restore the original secret image quality from the shares, which goes beyond the decoding with the bare human eye, was described in [15].

Recently, a concept of coding the images with shares that have some properties of randomness was proposed in [23]. It was an extension of the concept of Naor and Shamir made in such a way that in the case of black-and-white two-level images the shares have a truly random structure, as demonstrated in [24]. In the case of color images, this method made it possible only to provide for the randomness of color components in the shares, but not the true randomness of shares as whole images, as shown in [22]. Further, a method was proposed in [2] to improve the visual appearance of ran-

domness of the coding images, but it was evident that the resulting binary representations could not pass the randomness tests successfully.

In this paper the original contributions are as follows. First, the method is presented to make the coding images as close to random as possible. This can be achieved with very little additional loss of quality of the reconstructed image, beyond that introduced by the coding itself. Second, the randomness is extensively tested with the NIST Statistical Tests Suite [1], with a partial success. According to our best knowledge, in the literature there are no mentions on performing such tests for the shares in visual cryptography. Third, a relatively simple method of improving the quality of the decoded image by processing it numerically is presented. It improves the brightness and contrast of the decoded image. As a side effect, it reveals the degree to which the random processes present in the coding algorithm distort this image, with respect to the dithered version of the original secret. Differently from the method proposed in [15], where the shares were processed numerically, in our method only the decoded image is processed in the reconstruction algorithm.

Distracting the attention of observers from the fact that secret information is transmitted increases the security of the transmission process. Randomness of the data stream goes in a similar direction as making it seemingly contain unimportant information, as in the case of visual cryptography where the shares appear to contain some irrelevant images [19,27], or in the coding of audio signals which mimic the sounds emitted by animals [28].

The remaining part of this paper is organized as follows. In the next Section the general methodology used in visual cryptography is recapitulated. The coding of binary images, with the classic as well as the random method, is described in Section 2.1. The transformation of a color image into an image containing only the black, red, green and blue pixels, in which binary coding is possible, is presented in Section 2.2. The method of mixing the pixels which improves the visual appearance of the images is reminded in Section 2.3. Two methods of improving the randomness of the shares are presented in Section 4. The algorithms are illustrated with one artificial and one realistic example in Sections 3 and 5. The results of testing the randomness of the shares are presented in Section 6. Section 7 contains some remarks on problems related to the quality of decoded images and the possibility of improving it by simple numerical processing of the decoded image, and finally to the significance of color in visual cryptography. In Section 8 the paper is concluded.

The sources of the images shown in Figs. 3, 4, 5, 6, 9, 10 and 11 are available from [4].

Fig. 1 All possible 2×2 tiles and their indexes.

2 The coding method

Let us very shortly remind the basic notions and methods used in purely visual cryptography. Broader descriptions can be found in [2,22,23,24], with [22] being the most complete description.

The encoded image is called the *secret*. For the beginning, let us consider a two-level, black-and-white image. The secret is encoded in two images called the *shares*. The shares are printed on a transparent medium. The *decoding* consists in precisely overlaying the shares on each other, which makes the secret visible to a naked human eye.

2.1 Coding a two-level image

A single pixel in the secret is represented with a square of $n\times n$ pixels, called the *tile*, in each share. Let us assume $n = 2$. There are 16 different possible tiles shown in Fig. 1.

In the classic coding [20,21], in one share, called the *basic share*, the tile corresponding to each pixel of the secret is represented with one of the tiles 4, 6, 7, 10, 11, 13, chosen at random. The coding is done as follows. If the pixel in the secret is black, then in the other share, called the *coding share*, the negated tile is set (for tile 7 this would be tile 10). When the shares are overlaid, the tile corresponding to this pixel appears black. If the pixel in the secret is white, then in the coding share the same tile as in the basic share is taken. Then, after superposition, this tile is half-white. The contrast of the decoded image is half of that of the secret, but the pattern can still be easily seen.

This method was extended in [23] so that the shares became truly random, by that the basic share is drawn at random from among all the 16 possible tiles. This causes differences with respect to the classic decoding. The white pixels can become not only $1/2$ white, like in the classic method, but also $1/4$ or $3/4$ white (there are no errors in black pixels). The presence of such errors, analyzed in [24], is the cost of the true randomness of the shares.

2.2 Dithering a color image into two levels

To code a gray-level image it can be dithered into the black-and-white one. Similarly, the color image can be dithered into the one containing only black, red, green and blue pixels. Assume that an image is represented with a set of color columns, circularly R, G and B. Let us now represent each pixel in the secret with a 3×3 segment. In each color column, the pixels can be either color, or black. Four of such

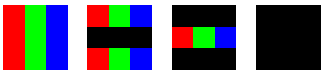

Fig. 2 Variants of a 3×3 pixel segment which encodes a single color pixel (in this case, white, gray, or black). (**a**) Full brightness – white pixel; (**b**) brightness $2/3$ – bright gray; (**c**) brightness $1/3$ – dark gray; (**d**) brightness 0 – black pixel. Image used in [2,22].

possible segments are shown in Fig. 2. Using the square segments is a matter of convenience, so the dimensions of the segment are 3×3. This makes it possible to represent four brightnesses of each color: 0, $1/3$, $2/3$ and $3/3$, by setting 0, 1, 2 or 3 pixels to color in a column. Three four-level colors form a 64-color palette, in which the secret can be dithered.

In the resulting image the pixel can be either black or full color: red, green or blue. Such an image can be coded, with either the classic or random coding, as described above. Each pixel of the secret is first replaced with a 3×3 segment, and then, each pixel of this segment is replaced by a 2×2 tile. Hence, a pixel is expanded into $3^2\times2^2=36$ pixels.

2.3 Mixing

The structure of the segment shown in Fig. 2 implies that the R, G and B pixels in the shares and in the decoded image are organized in columns. If the coding is random, their values are random, but their locations remain fixed to color columns [22]. Later it was proposed in [2] to mix the pixels, in two ways: by the 4×4 tiles, and by pixels, where all the 36 pixels within each segment are mixed without observing the tile structure. Here, we shall use only the mixing by pixels, because the resulting decoded images have a better visual appearance [2]. In this paper, only mixing the randomly coded images is shown; however, the same can be done also for classic coding.

From the method of coding with tiles of Fig. 1 it follows that in each share, there is one black pixel per one color pixel; hence, the number of black pixels equals the sum of numbers of color pixels. This holds strictly in classic coding and in the sense of the average in random coding. Therefore, such an image structure is far from being random.

3 Test example

A simple test image (used also in [2,22]) and its coding with the described methods is shown in Fig. 3. It contains squares in basic and complementary colors, and four shades of gray, which happen to be accurately represented in the palette used. Pixels are replaced by 3×3 segments, with proper pixels set *on*, which forms the decomposed image in Fig. 3b. The result of decoding from the classical coding is shown in Fig. 3c, and from the random coding in d. The mixing makes this image lose the striped texture, Fig. 3e. One of two shares for images d and e are shown in Figs. 3g, h (images i, j will be referred to later on).

Some of the stages of the coding and decoding process reduce the quality of the images. The first reason for quality loss is the dithering into the 64-color palette. The second reason is the decomposition of the dithered image into the two-level image. In the white pixel of the secret all the pixels of the corresponding segment are on, while in the red (green, blue) pixel only the pixels in the red (green, blue) column are on. Inevitably, in this representation the image is darker than the original. The coding by decomposing into tiles (image c) further reduces brightness and contrast, which is typical in the majority of purely visual cryptographic methods. The random coding (**d**) additionally introduces its specific errors. The mixing (e) removes the column-wise structure of the image which improves its evenness, but amplifies the granularity of texture.

4 Improving the randomness

As written before, half of the pixels in a share are black, and the remaining ones are R, G and B. Note that before the mixing is performed, in each column the pixels are coded with tiles (Fig. 1) which are half-black and half-color, according to the color of the given column. Consider the segment 6×6. From 36 pixels, 18 are black, and the remaining 18 are: 6 R, 6 G and 6 B. To make the numbers of pixels in the colors equal, 3 black pixels should be turned to red, 3 to green and 3 to blue. Then, in the segment there would be 9 black, 9 red, 9 green, and 9 blue pixels. In the whole image, $\frac{3}{36}=\frac{1}{12}$ of black pixels should be modified by changing to red, $\frac{1}{12}$ to green, etc. Generally, the black pixels to be changed should be drawn at random, without any reference to the contents of the secret image, which should ensure the randomness of the shares.

Free method The totally random method of modifying the black pixels into color ones consists in changing the proper numbers of black pixels totally at random, separately in each share. Locations of pixels to be modified as well as their colors are drawn at random. In the decoding, therefore, some black pixels turn into color if their colors occur equal (rare case), black if one of the pixels remained black, and black if pixels are in different colors (two different ideal filters in different basic colors overlayed give black, or very dark in non-ideal case). In this way, the decoding errors, additional to those resulting from the random coding by the tiles, appear in the decoded image. As it will be shown in the examples in the next section, these errors, although visible, do not make it impossible to recognize the shapes and colors in the decoded images. Adding the equal number of randomly displaced R, G and B pixels corresponds to adding some whiteness to the image, so this decreases the contrast of the

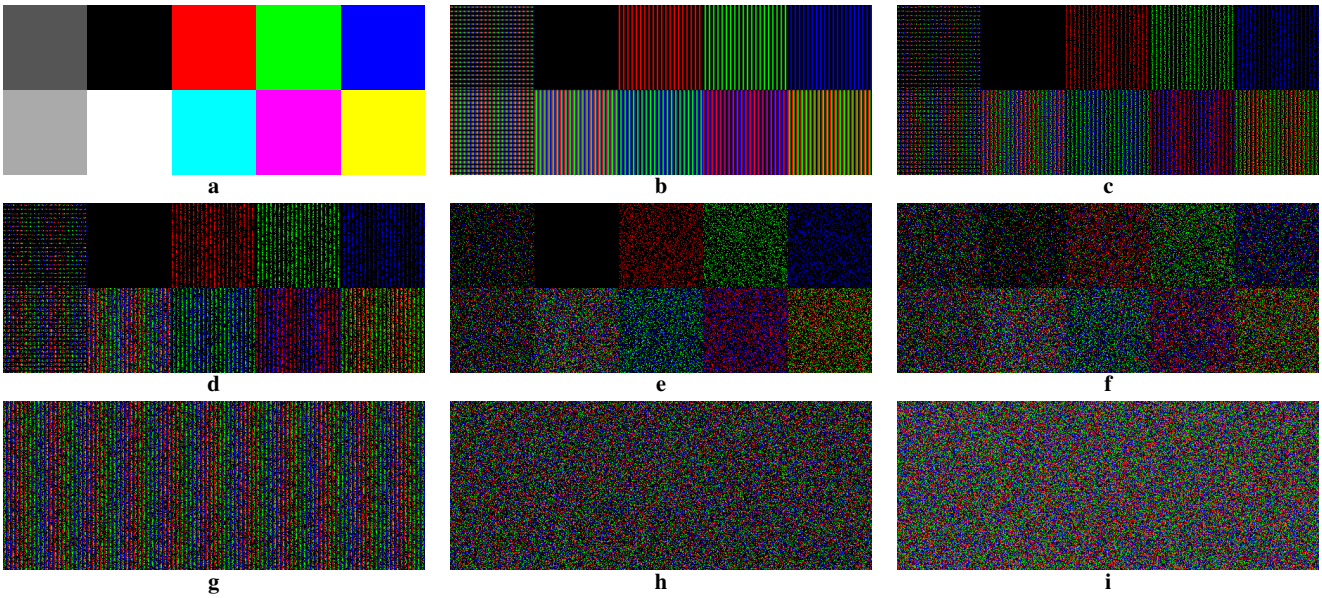

Fig. 3 Illustration of coding and decoding for a simple test image. (**a**) Secret, 100×40; (**b**) image **a** dithered and decomposed into color stripes for coding, 300×120; all further images are 600×240: (**c**) coded classically; (**d**) coded randomly, no mixing; (**e**) coded randomly, mixed; (**f**) as **e**, modified with *free* method; (**g**) one share of **d**; (**h**) one share of **e**; (**i**) one share of **f**. The same or similar example of a secret image **a** was used in [2,22]. The sources of the images shown in this and the following images are available from [4].

result. This method will be denoted as *free* or F. The result is illustrated in Fig. 3f and i.

Intermediate method Optimally, adding color pixels to the shares should not introduce any changes in the decoded image. In cases when the given pixel is black in both shares, then in only one share this pixel is changed into a random color. However, from the coding algorithm it follows that the overlapping black pixels appear more frequently in brighter regions of the secret. Modifying only these pixels and leaving the remaining ones untouched would reveal the shade of the secret in the share. Therefore, the changes should be made also in the pixels where a black pixel in one share overlaps a color pixel in the other share. In such a case, the black pixel can be changed to a random color, but different from the color of the corresponding pixel in the other share. Pixels in which there is a same color in both shares can not be changed. This method will be denoted as *intermediate* or I.

This procedure can be performed globally (*global intermediate* method): a pixel is drawn from the image at random, and if at least in one share the pixel is black, the modification is made; this is done until the proper number of black pixels are changed into each color. The resulting decoded image is strictly equal to that generated from the shares without modification (Fig. 3e). However, a problem appears: a shadow of the secret is visible in the shares, as shown in Fig. 4. The reason for this is that the pixels are randomly drawn for changes, but the probability of meeting a double black or single black pixel in the shares is not independent of the contents of the image; hence, the density of changes actually made is far from constant throughout the image. Similar problems, related to contrast change in the image, where the CMY color space was used, were reported in [11] to have appeared in the earlier work of this author [12], cited in that paper, and were overcome in [11].

To correct this clear drawback of the *global intermediate* method, it is proposed to perform the changes *locally*, so that the number of color and black pixels is balanced in each 6×6 segment, in each share. Hence, the same is done as described in the *intermediate* method, but in each segment separately. Further, when we refer to the *intermediate* method, or I, without any qualifier, we always have in mind its local version.

Coordinated method Unfortunately, the mutual locations and colors of pixels do not always make it possible to perform all the necessary nine changes in two shares, in each segment. The number of missing changes in the test image is shown in Fig. 4c. Note that the largest numbers of missing changes are found in the brightest region of the image.

There is no other way than to complement these missing changes with the proper number of random changes, like in the *free* method. This introduces errors in the decoded image, but their number is as small as possible, because these changes are made after all other possibilities are used. In practice, it was measured that the number of such error introducing changes is less than 2% of all changes made in a small (`test100`) and less than 0.5% in a large one (`peppers`). It must be stated that in some tiles it happens that there are no black pixels left to be changed; such cases constitute less than 0.1% in small images and less than

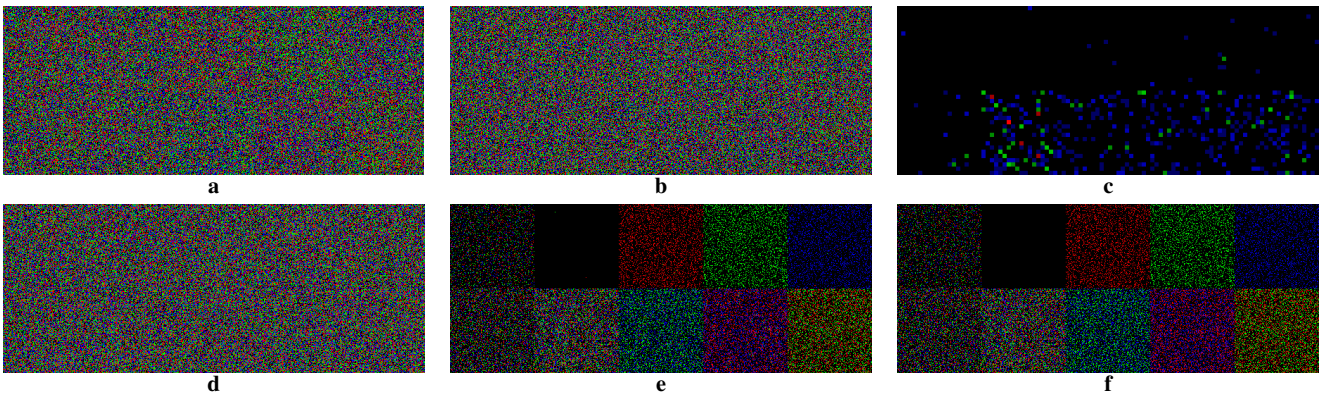

Fig. 4 Illustration of information leakage and the remedy for it for a simple test image. (**a**) Share from global intermediate method, shadow of the secret revealed; (**b**) share from local intermediate method, shadow absent; (**c**) numbers of missing changes in segments, in one share (colors indicate numbers from 0 to 6: black, dark blue, blue, dark green, green, dark red, red). (**d**) Share from coordinated method; (**e**) decoded image from coordinated method: some errors visible; (**f**) decoded from random method with mixing (Fig. 3e), for comparison.

0.01% in large ones, so we shall consider this problem as negligible.

The described *local intermediate method* corrected with the addition of *free* elements will be called the *coordinated* method or C, due to that the changes are coordinated between the shares in one segment.

5 Example of a natural image

The image `parrots` from [9] was chosen due to its bright and varied colors. Its size was reduced by resampling, to show better the coded images which have an increased resolution, Fig. 5. In this example, not only the color evenness, but also the preservation of details can be assessed. Despite the losses of quality, in the image decoded from random shares the objects can be recognized and the colors can still be noticed. The additional errors introduced by the *coordinated* method, the most frequent in the bright regions of the image, do not reduce its quality significantly; rather, they tend to slightly improve its contrast.

6 Testing the randomness

In several places in this paper we used the term *true randomness*. When doing this we have implicitly made a simplifying assumption that *"the sequence which possesses all the properties that a truly random sequence would have"* can be considered as *"sufficiently random"*. The cited statement comes from the abstract of [14]. The author of this phrase extends this line of thought by stating that a satisfactory testing *"would require an infinite number of tests"*. By referring to an infinite numer of tests these considerations become more philosophical than technically orientated. However, although far from infinite, the sets of tests have been designed. Among these, the `TestU01` library [17] and the NIST Statistical Tests Suite [1] are easily accessible and frequently used by cryptographers ([17]: over 1000 citations; [1]: over 2800 citations [10]). The NIST Suite is more recent and seems to be easier to use. Therefore, we have chosen it as a practical solution to the question of how to check whether the data we generate can be considered as random or not.

While being aware of that the *true randomness* is generally a difficult and still open problem, we have treated the use of the NIST Suite as a means to classify the sequences of bits generated by our methods as only *apparently unordered* but definitely *not random*, or *possible to be considered as random*. Seeking to apply an infinite number of tests would not make us closer to the answer to our basic question: yes or not. So, if there are no indications to treat a sequence as not random, we shall consider it not only *looking as random*, but *truly random*, all the time in a limited, engineering sense.

For the tests, a pixel was expressed as two bits: 00 for black, 01 for red, 10 for green and 11 for blue. This arbitrary assignment was believed not to influence the result. Pixels were saved in two separate files: by rows and by columns. Both shares were tested. Therefore, for each image there were four data files. Besides the above mentioned two images, a number of well known benchmark images were used: `baboon`, `peppers` and `Lena` (to be found in the sources cited in the historical web site [13]).

Default values of the parameters of the NIST software were used; in particular, $\alpha = 0.01$. Results for the tests which have subtests (`CumulativeSums`, `NonOverlappingTemplate`, `RandomExcursions`, `RandomExcursionsVariant`) were shown together, so finally 188 tests with their subtests were treated as just 15 tests.

To make it possible to capture the state of the randomness for one image as a whole, it has been attempted to show the results for 100 random realizations of coding, two directions and two shares, in one page. This conforms with the concept of *small multiples* introduced by Tufte [26], which

Fig. 5 Illustration of decoding for a natural image `parrots`. (**a**) Secret, 384×256; (**b**) image **a** dithered and decomposed into color stripes for coding, 1152×768; all further images are 2304×1536: (**c**) coded classically; (**d**) coded randomly, mixed; (**e**) modified with the *free* method; (**f**) modified with the *coordinated* method.

advices to present all the relevant data together, so that they can be perceived simultaneously.

The results are presented in the form of histograms of p-values (Figs. 7, 8). Further we shall refer to the p-values as simply p. The width of the bins is $\alpha/4$. The counts for the test failures, $p \leq \alpha$, denoted in the key of the graphs with a graphical symbol as *low*, are shown in shades of red. The histogram values for the success of the tests, denoted as *good*, are shown in the shades of grey. Separately, to the right of the range of p, the number of cases for which the preconditions for the tests were not met are shown with the shades of blue and denoted with a question mark meaning *not applicable*. This can happen for tests: `RandomExcursions`, `RandomExcursionsVariant`, and `Universal`, for which the NIST Tests Suite can issue warnings. The data for the *basic shares*, denoted as share 1, are shown with full sym-

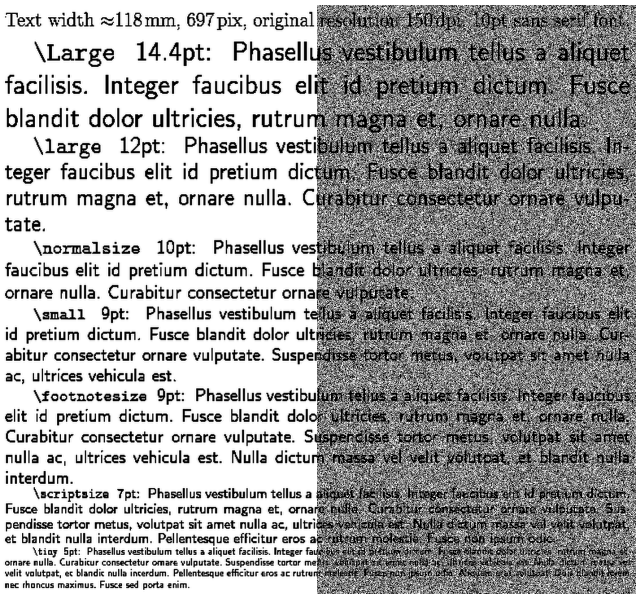

Fig. 6 Image `TextBlack` (left, half) and the result of its decoding from the random coding (right, half).

bols, and for the *coding shares* – share 2, with empty symbols. The data for reading the pixels by columns, vertically, are shown with bars, and by rows, horizontally, with circles.

To show clearly the important data for $p \le \alpha$ which indicate the failures of the tests, a nonlinear axis for p is used. The transformation is $p \to p^a$, with a such that the point 0.01 takes the place of the former point 0.1.

Let us start with the results of analysis for image `TextBlack` shown in Fig. 6 used in [24] to illustrate the basic random algorithm for two-level images, introduced in [23]. The image and the result of decoding are shown together.

From the graphs of p-values shown in Fig. 7 it results that the basic random version of the coding is truly random. For all the tests, the majority of realizations passed successfully. The tests that failed are in minority, and their frequencies conform to the generally flat distribution of p-values. The cases which did not meet the initial conditions of some tests are also rare.

The results for the image `parrots` (shown in Fig. 5a), coded with the *coordinated* method, are shown in Fig. 8. The same results for this one and other images are shown in an abstracted form in Tab. 1, where for each test only the numbers of *low*, *good* and *not applicable* results are given. This hides the distribution of p-values but makes it possible to present the results for many images condensed in a limited space. The histograms themselves can be found by the interested reader in the supplementary material [3].

In three images the numbers of black pixels were reduced with three methods: *free*, *intermediate* and *coordinated*. Results from the coordinated method are generally worse than those of the free method. This indicates that the changes in the shares restricted to operating on each segment separately, although giving visually better results, is strongly less random than making the changes freely. The *free* method, however, has an important drawback of reducing the contrast of the decoded image, seen in images of Figs. 4f and 5e. It does not guarantee the randomness in all the cases (`baboon`: `ApproximateEntropy`, `parrots`: `Runs`, `LongestRun`, `ApproximateEntropy`, `Serial`). The local *intermediate* method gave less random results than the *coordinated* one, especially for the tests `Frequency` and `CumulativeSums`. Therefore, we discontinued testing this method, in spite of that this method yields the decoded images free from additional errors. We did not show the results for the global *intermediate* method, which is unacceptable due to information leaks, but unexpectedly, it generally has a better randomness (some detailed results are available in the supplementary material [3]).

The tests which can be considered as confirming the randomness of the color shares are `Frequency`, `BlockFrequency`, `CumulativeSums`, `Rank`, `RandomExcursions`, `RandomExcursionsVariant` and `LinearComplexity`.

The tests which always or nearly always reject the randomness are `Runs`, `LongestRun`, `OverlappingTemplates` and `ApproximateEntropy`.

It can be also noted that the preconditions for `RandomExcursions` and `RandomExcursionsVariant` were not met over twice more frequently for images read horizontally than for those read vertically (Fig. 8), which is an indication of directionality, present in the expectedly isotropic objects.

Finally, it should be stated that the search for true randomness, successful for the black-and-white images, is still an unreachable target for the color ones.

7 Remarks on quality and color

7.1 Hiding the pixels and the quality of decoded images

The vital advantage of the visual cryptography methods is the possibility to decrypt the secret image from the shares with the bare eye only, without any equipment. In this process, the ability of the human eye to see the objects even if they are distorted due to noise or errors is important. This ability was helpful in reading the decoded images, as the errors are inevitably inherited in the random methods used. Besides these errors, the visual cryptography methods heavily rely on the process of hiding some pixels in one share by other pixels in another share. In this way, only (or nearly only) those pixels are exposed to the human observer which are necessary to form the useful image, while all (or nearly all) the remaining ones are hidden. Hiding the pixels which are not intended to be seen plays the fundamental role in the decoding process. Therefore, it seems not possible to get

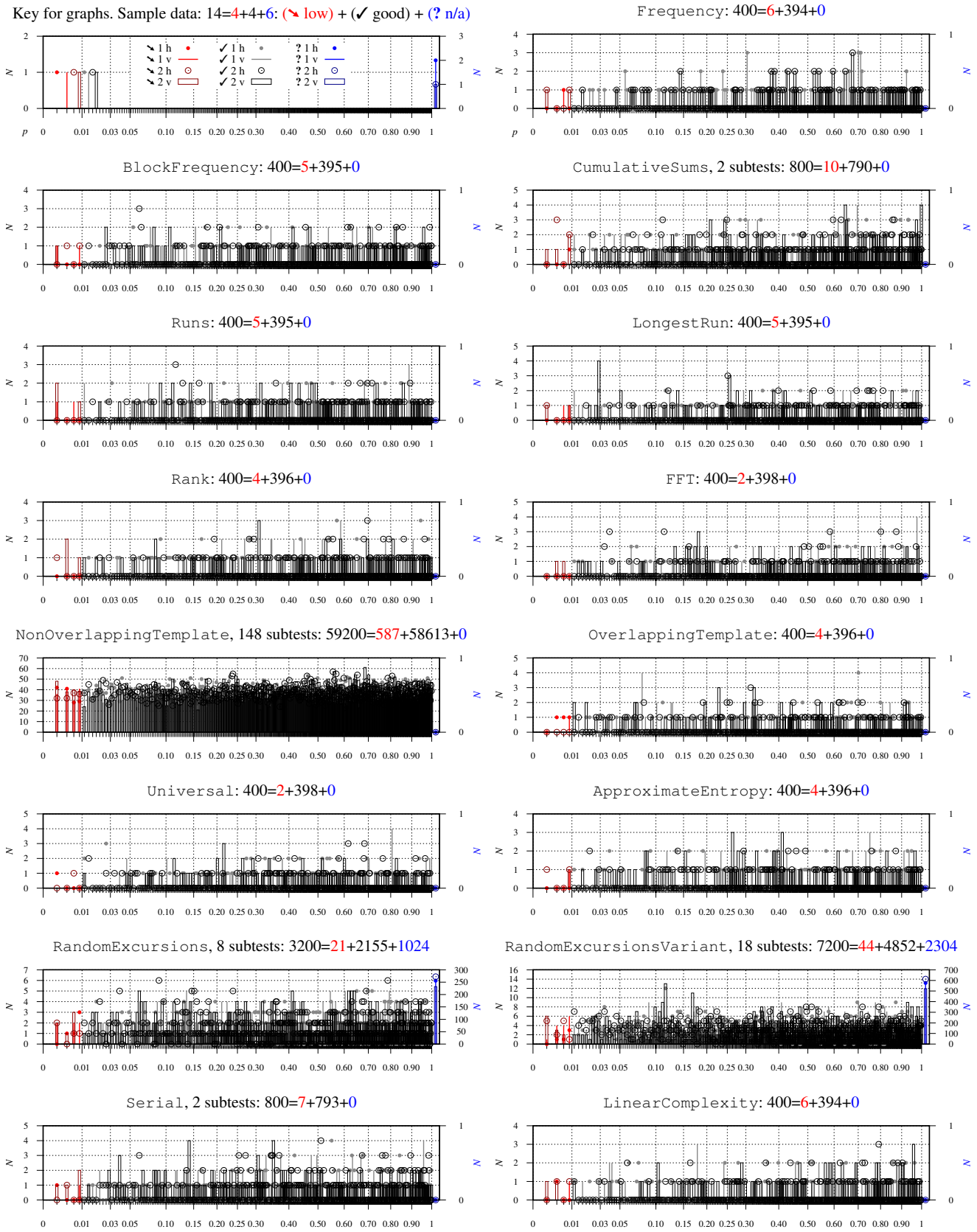

Fig. 7 Histograms of p for two-level, black-and-white image `TextBlack`. Key is shown in the upper left sub-image, containing a histogram for sample data.

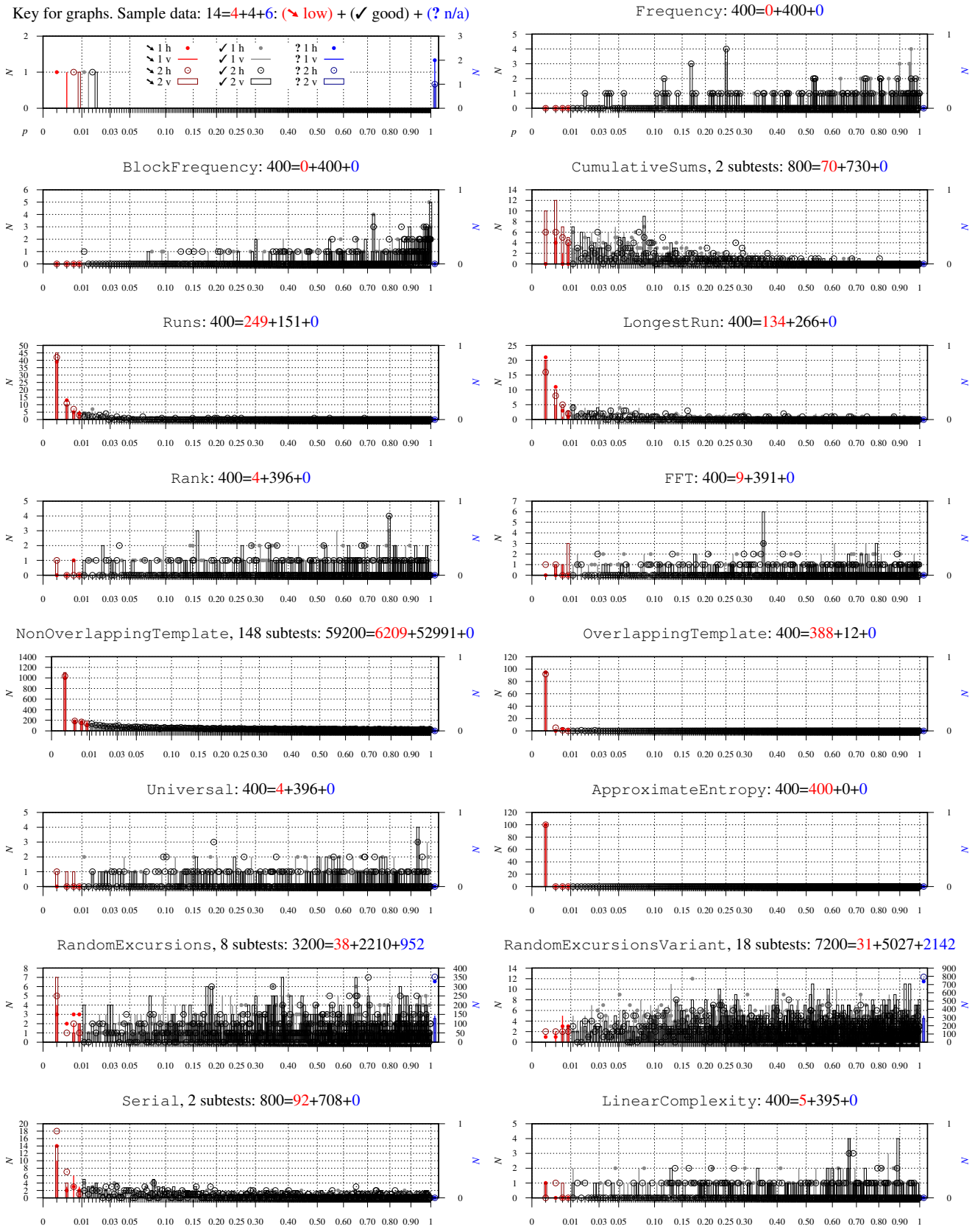

Fig. 8 Histograms of p for color image `parrots`. Key is shown in the upper left sub-image.

Table 1 Abstracted results of statistical randomness tests shown as pairs (or triplets, where applicable) of numbers of realizations, with color text: (*low good n/a*). Letters F, I, C after names of color images denote the modification of shares into random form with methods: *free*, *intermediate* and *coordinated*, respectively.

Image	size	Frequency	Block Frequency	Cumulative Sums	Runs	LongestRun	Rank	FFT	Non Overlapping Template	Overlapping Template	Universal	Approximate Entropy	Random Excursions	Random Excursions Variant	Serial	Linear Complexity
TextBlack	709×656	6 394	5 395	10 790	5 395	5 395	4 396	2 398	587 58613	4 396	2 398 0	4 396	21 2155 1024	44 4852 2304	7 793	6 394
test100 F		2 398	0 400	3 797	4 396	30 370	4 396	10 390	815 58385	17 383	0 0 400	24 376	10 1358 1832	16 3062 4122	14 786	8 392
test100 I	100×40	354 46	6 394	709 91	339 61	286 114	5 395	14 386	2755 56445	179 221	0 0 400	306 94	4 460 2736	2 1042 6156	19 781	3 397
test100 C		6 394	4 396	12 788	122 278	164 236	6 394	14 386	20300 57170	119 281	0 0 400	276 124	15 1377 1808	15 3117 4068	17 783	5 395
baboon F		0 400	0 400	0 800	39 361	35 365	5 395	4 396	2016 57184	153 247	7 393 0	253 147	25 2127 1048	40 4802 2358	21 779	2 398
baboon I	125×120	20 380	0 400	37 763	398 2	206 194	4 396	54 346	7397 51803	399 1	22 378 0	400 0	17 1727 1456	42 3882 3276	119 681	2 398
baboon C		0 400	2 398	2 798	390 10	207 193	7 393	40 360	7592 51608	395 5	15 385 0	400 0	13 2139 1048	32 4810 2358	151 649	3 397
parrots F		4 396	0 400	7 793	352 48	352 48	5 395	20 381	9440 49760	400 0	6 394 0	400 0	41 2863 296	43 6491 666	318 482	4 396
parrots I	384×256	398 2	2 398	796 4	400 0	400 0	3 397	327 73	25861 33339	400 0	77 323 0	400 0	27 1757 1416	21 3993 3186	402 388	8 392
parrots C		2 398	3 397	4 796	400 0	400 0	2 398	327 73	24389 34811	400 0	75 325 0	400 0	30 2834 336	96 6348 756	404 386	6 394
peppers C	512×512	0 400	3 397	2 798	400 0	400 0	3 397	399 1	35461 23739	400 0	296 104 0	400 0	38 2954 208	88 6644 468	405 395	4 396
Lena C	512×512	2 398	0 400	4 796	400 0	400 0	2 398	399 1	35429 23711	400 0	284 116 0	400 0	32 3000 168	76 6746 378	404 396	3 397

rid of the reduction of brightness and of other negative results of the hiding process, as far as the RGB color space is used (we have not explored the application of the CMY space in which the colors arise in the process of overlapping the shares).

7.2 Reconstruction of the decoded image with numerical processing

The possibility of numerical processing of the reconstructed secret (independently of the decoding with a bare eye) opens new possibilities. This is somewhat remote from our main domain of interest, which is the encryption itself, but let us explore this path of reasoning to some extent.

Let us remind that in the dithering process, the following four intensities are assigned to the R, G and B components: 0, $1/3$, $2/3$, 1. If there are no errors, these intensities are reflected by the presence of 0, 2, 4, 6 pixels in R, G and B, respectively. So, the numbers of pixels in the range $[0,6]$ can be projected into the range of color intensities $[0,255]$. The odd numbers of pixels are errors, but when only the reconstructed secret is available (or shares), the information on the nature of errors is lost. Therefore, the odd values can be assigned to the nearest even value up or down, at random, or simply left unchanged – this second approach was used here. In this way, the decoded image can be restored back to the state close to that of the image dithered into the 64-color palette. In the case of the test image of Fig. 3**a** the dithering introduces no distortion at all (loss of quality visible in Fig. 3**b** is due to decomposition into color stripes). The result of restoration for this image and for the image of Fig. 5, in comparison to the dithered images, are shown in Fig. 9. The reconstructed images reveal in a visual way what is the level of errors introduced at the stages of the coding process. These errors result from using the random coding with all the tiles of Fig. 1 and of balancing the colors in the shares in the *coordinated* method. It can be seen that besides the presence of the errors, a considerable part of information is maintained.

It can be postulated that the raw decoded image is what can be used in the field conditions, while in the laboratory conditions the image can be further processed to receive the reconstructed image.

7.3 Why color?

Finally, let us consider the question of color in visual cryptography, having in mind that some loss of quality is unavoidable and hence the color information is attenuated. It is not necessary to explain that color is a very important feature used by the visual system of humans. It simply follows from our everyday experience. When poor light conditions make the color part of our visual system (photopic vision) switch off, we feel an important decrease of our visual abilities, although our grey-level vision (scotopic vision) works properly with dim light. Furthermore, to justify the existence of research on color visual cryptography it is not enough to state that such research is actually being carried out, which can be deducted from the large number of publications emerging each year.

One of the examples of visual cryptography for images which are only black-and-white is the coding of images with text. This example is used throughout the literature, for example in [7] (we went the same way in some of our previous papers). After decoding, the text can be read with smaller or greater ease. The loss of quality inherent in the methods makes it doubtful whether the color images can be effectively used after decoding. This doubt results mainly from that the color images are considered as suitable for displaying complex objects, while in simple-shaped objects color

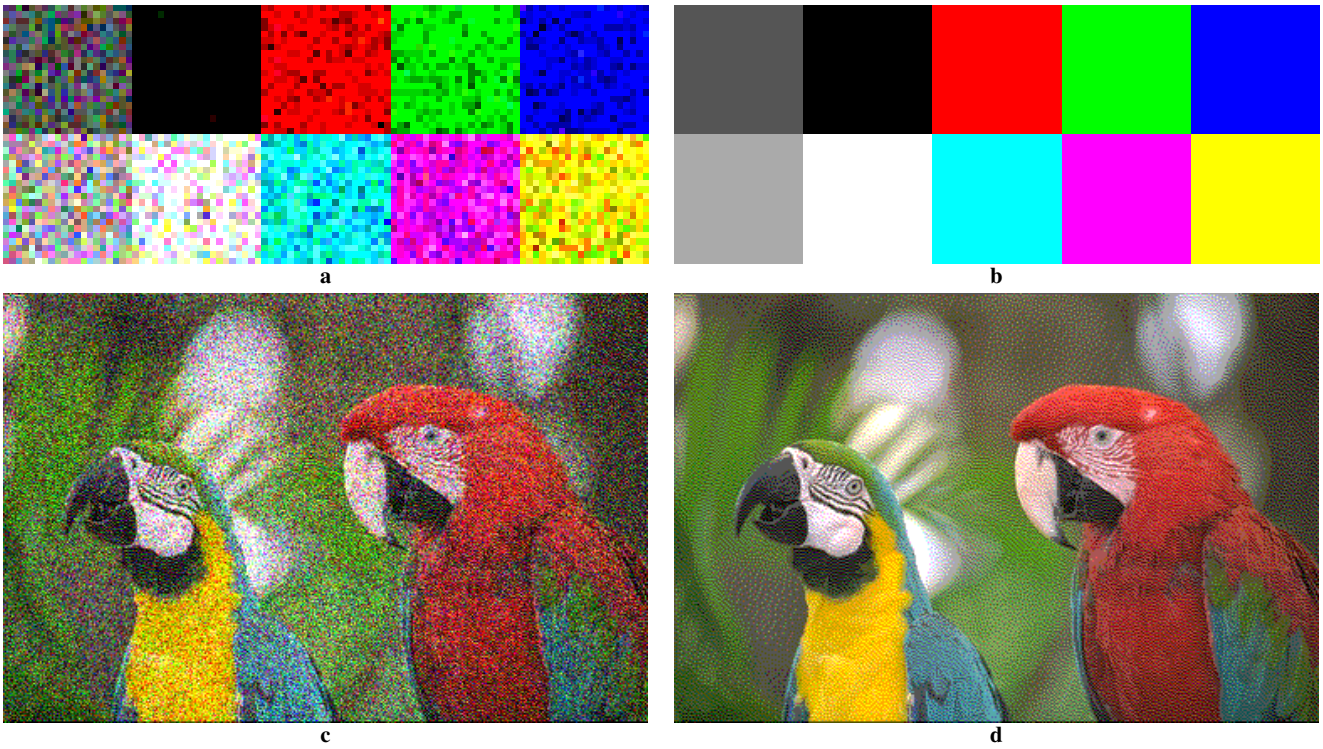

Fig. 9 Illustration of reconstruction for a test and a natural image in comparison to the versions dithered in the 64-color palette used for color coding. (**a**) Image reconstructed from the image of Fig. 3**e**; (**b**) image of Fig. 3**a** dithered. (**c**) Image reconstructed from the image of Fig. 5**f**; (**d**) image of Fig. 5**a** dithered.

is of secondary importance. Considering the image of parrots used throughout this paper seems to be of little practical importance, as in the decoded image the major part of the natural beauty and richness of this image is lost.

However, color is widely used and is very important also in simple images, which can be recognized easily. Let us consider the flags of countries, which are usually composed of very simple shapes. Even if some small graphical elements are present in them, it is rather the general shape and colors which make the flags recognizable. As an example of the significance of color let us look at the selected flags shown in Fig. 10. Besides that the shapes are radically simple, without color information the significance of the flags would remain unknown.

The images to be coded can even be well known to the reader beforehand, like in the case of the emergency icons (Fig. 11a). They can be recognized at a glance, even in the low-quality decoded images, largely due to color information, which is chosen according to the meaning of the icon.

Hence, it seems reasonable that color visual cryptography can be applied, for example, in cases when some simple iconic information of high importance should be secretly and safely transmitted to a place where there are no technical resources available to decrypt it.

8 Conclusions and perspective

The concept of visual coding with shares in which colors are coded in the completely random way was modified by locally mixing the contents of the shares and by equalizing the numbers of black, red, green and blue pixels in the shares. For this, a specified number of black pixels in the shares were changed into color ones. The developed method of modification did not reduce the quality of the decoded image significantly with respect to that of the previously known methods. The battery of 15 NIST randomness tests was applied to experimentally check the randomness of the shares for a set of one two-level image and five benchmark color images. From this battery, all the tests passed with a success for the randomly coded two-level image. For color images, seven tests passed with success, four failed nearly always, and the remaining four ones gave varied results. Therefore, the goal of hiding the process of information transfer in the purely visual color cryptography is far from being reached, although the presence of successes as well as failures indicate that some level of randomness has been attained.

It has been shown how the quality of the decoded images can be improved with simple numerical calculations made solely on the decoded image. There is no need to analyze the shares.

Further studies should be concentrated upon explaining the reasons for failures and on improving the coding al-

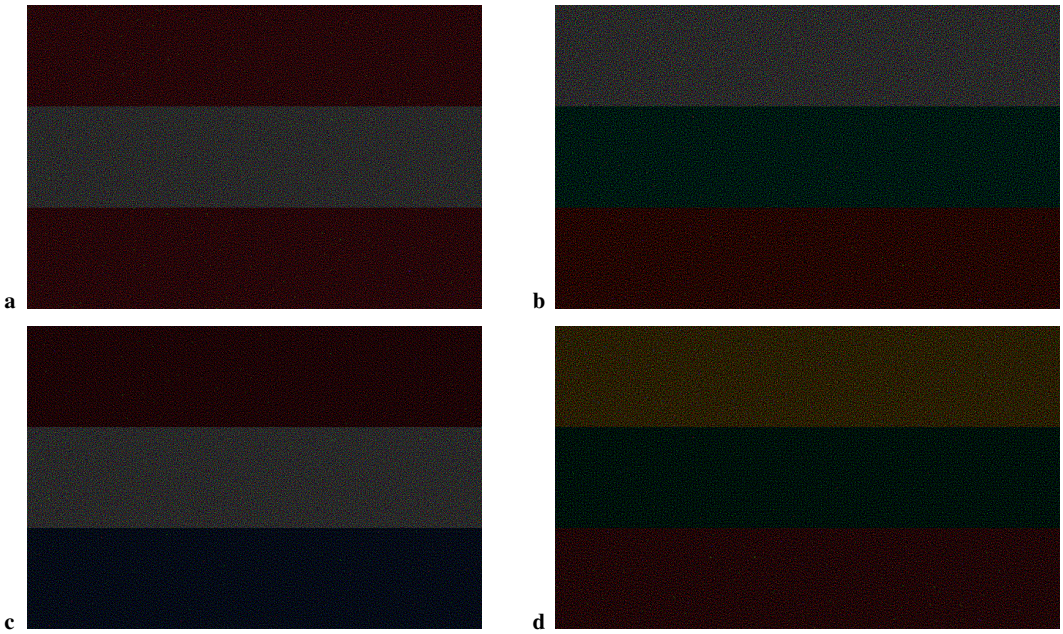

Fig. 10 Examples of iconic information where color matters: flags; decoded images. (**a**) Austria; (**b**) Bulgaria; (**c**) Netherlands; (**e**) Lithuania. Source for flags: [16].

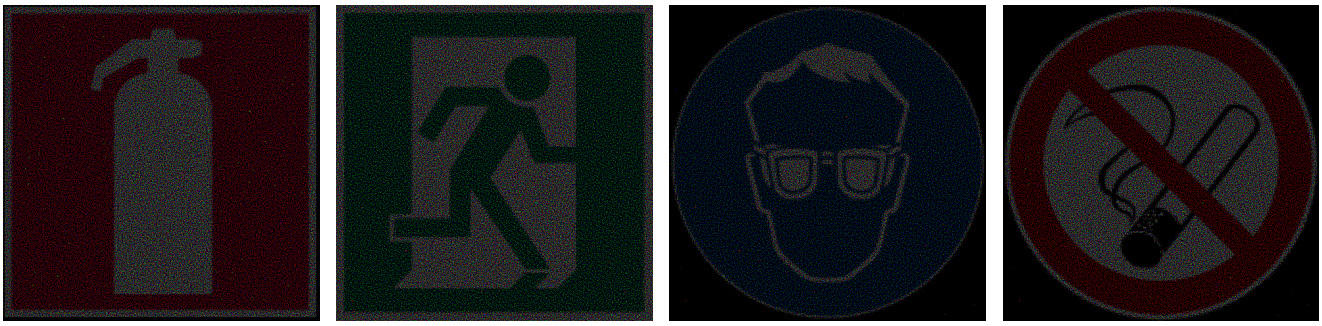

Fig. 11 Examples of emergency icons where color matters; decoded images.

gorithms to make more statistical randomness tests pass. It seems that one of the directions could be breaking the requirement of local balance between colors in the shares.

Conflict of interest

The authors declare that they have no conflict of interest.

Acknowledgements We would like to thank the Reviewers of this paper for their insightful remarks which helped us to improve and extend the contents of the paper.

References

1. Bassham, L.E., Rukhin, A.L., Soto, J., et al.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD, USA (2010). URL https://www.nist.gov/manuscript-publication-search.cfm?pub_id=906762. Series: Special Publication (NIST SP), Rep. No. 800-22 Rev 1a
2. Chmielewski, L.J., Gawdzik, G., Orłowski, A.: Towards color visual cryptography with completely random shares. In: Proc. Conf. PP-RAI 2019 – Konf. Polskiego Porozumienia na rzecz Rozwoju Sztucznej Inteligencji, pp. 150–155. Wrocław University of Science and Technology, Wrocław, Poland (2019). URL http://pp-rai.pwr.edu.pl/PPRAI19_proceedings.pdf
3. Chmielewski, L.J., Nieniewski, M., Orłowski, A.: Histograms – supplementary material to: Testing the randomness of shares in color visual cryptography (2021). Submitted as supplementary material together with this paper.
4. Chmielewski, L.J., Nieniewski, M., Orłowski, A.: Visual cryptography – file repository (2021). URL http://www.lchmiel.pl/visualcrypto. [Accessed Jun 2021]
5. Cimato, S., Yang, C.N.: Visual Cryptography and Secret Image Sharing (Digital Imaging and Computer Vision), 1st edn. CRC Press, Inc., Boca Raton, FL, USA (2012). DOI 10.1201/b11068
6. Dahat, A.V., Chavan, P.V.: Secret sharing based visual cryptography scheme using CMY color space. Procedia Computer Science **78**, 563–570 (2016). DOI 10.1016/j.procs.2016.02.103. 1st International Conference on Information Security & Privacy 2015
7. De Prisco, R., De Santis, A.: Color visual cryptography schemes for black and white secret images. Theoretical Computer Science **510**, 62–86 (2013). DOI 10.1016/j.tcs.2013.09.005
8. Dhiman, K., Kasana, S.S.: Extended visual cryptography techniques for true color images. Computers & Electrical Engineering

70, 647–658 (2018). DOI 10.1016/j.compeleceng.2017.09.017
9. Franzen, R.: RWF's Eclectic Miscellany (2015). URL http://r0k.us. [Accessed Apr 2019]
10. Google: Google Scholar (2021). URL https://scholar.google.com. [Accessed Mar 2021]
11. Hou, Y.C.: Visual cryptography for color images **36**(7), 1619–1629 (2003). DOI 10.1016/S0031-3203(02)00258-3
12. Hou, Y.C., Chang, C.Y., Lin, F.: Visual cryptography for color images based on color decomposition. In: Proc. 5th Conf. Information Management, pp. 584–591. Taipei, Taiwan (1999)
13. Huber, D.: The Computer Vison Homepage (2004). URL https://www.cs.cmu.edu/~cil/vision.html. [Accessed Mar 2021]
14. James, F.: Chaos and randomness. Chaos, Solitons & Fractals **6**, 221–226 (1995). DOI 10.1016/0960-0779(95)80028-F
15. Jin, D., Yan, W., Kankanhalli, M.S.: Progressive color visual cryptography. Journal of Electronic Imaging **14**, 033,019 (2005). DOI 10.1117/1.1993625
16. Krmela, D.: Flagpedia.net. Flags of the World (2021). URL https://flagpedia.net. [Accessed Mar 2021]
17. L'Ecuyer, P., Simard, R.: TestU01: A C library for empirical testing of random number generators. ACM Transactions on Mathematical Software **33**(4) (2007). DOI 10.1145/1268776.1268777
18. Liu, F., Yan, W.Q.: Visual Cryptography for Image Processing and Security: Theory, Methods, and Applications. Springer International Publishing, Cham (2014). DOI 10.1007/978-3-319-09644-5
19. Lou, D.C., Chen, H.H., Wu, H.C., Tsai, C.S.: A novel authenticatable color visual secret sharing scheme using non-expanded meaningful shares. Displays **32**(3), 118–134 (2011). DOI 10.1016/j.displa.2011.02.001
20. Naor, M., Shamir, A.: Visual cryptography. In: A. De Santis (ed.) Advances in Cryptology — EUROCRYPT'94. Proc. Workshop on the Theory and Application of Cryptographic Techniques, *Lecture Notes in Computer Science*, vol. 950, pp. 1–12. Springer, Perugia, Italy (1995). DOI 10.1007/BFb0053419
21. Naor, M., Shamir, A.: Visual cryptography II: Improving the contrast via the cover base. In: M. Lomas (ed.) Security Protocols. Proc. Int. Workshop on Security Protocols, *Lecture Notes in Computer Science*, vol. 1189, pp. 197–202. Springer, Cambridge, United Kingdom (1997). DOI 10.1007/3-540-62494-5_18
22. Orłowski, A., Chmielewski, L.J.: Color visual cryptography with completely randomly coded colors. In: M. Vento, G. Percannella (eds.) Proc. Int. Conf. on Computer Analysis of Images and Patterns CAIP 2019, vol. 11678, pp. 589–599. Springer, Salerno, Italy (2019). DOI 10.1007/978-3-030-29888-3_48
23. Orłowski, A., Chmielewski, L.J.: Generalized visual cryptography scheme with completely random shares. In: N. Petkov, N. Strisciuglio, C.M. Travieso (eds.) Proc. 2nd Int. Conf. Applications of Intelligent Systems APPIS 2019, pp. 33:1–33:6. Association for Computing Machinery, Las Palmas de Gran Canaria, Spain (2019). DOI 10.1145/3309772.3309805
24. Orłowski, A., Chmielewski, L.J.: Randomness of shares versus quality of secret reconstruction in black-and-white visual cryptography. In: L. Rutkowski, et al. (eds.) Proc. Int. Conf. on Artificial Intelligence and Soft Computing ICAISC 2019, vol. 11509, pp. 58–69. Springer, Zakopane, Poland (2019). DOI 10.1007/978-3-030-20915-5_6
25. Thomas, S.A., Gharge, S.: Review on various visual cryptography schemes. In: Proc. 2017 Int. Conf. Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), pp. 1164–1167. Mysore, India (2017). DOI 10.1109/CTCEEC.2017.8455136
26. Tufte, E.R.: The Visual Display of Quantitative Information. Graphic Press, Cheshire, CT (1983)
27. Wu, H., Wang, H., Yu, R.: Color visual cryptography scheme using meaningful shares. In: Proc. 8th Int. Conf. Intelligent Systems Design and Applications ISDA 2008, vol. 3, pp. 173–178. Kaohsiung, Taiwan (2008). DOI 10.1109/ISDA.2008.130
28. Zuber, K.W., Opieliński, K.J.: Animal mimicry for covert communication with arbitrary output distribution: Beyond the assumption of ignorance. Vibrations in Physical Systems **30**(1), 2019,119 (2019). URL http://vibsys.put.poznan.pl/vibrations-in-physical-systems-vol-2019-30-1