STEGANOGRAPHY USED FOR COPYRIGHT PROTECTION IN MATLAB ENVIRONMENT

Robert Halenár[*]

University of Ss. Cyril and Methodius, Faculty of Mass Media Communication, Nám. J. Herdu 2, 91701 Trnava, Slovak Republic

(Received 16 June 2014, revised 27 August 2014)

Abstract

Problems regarding copyright protection have no universal solution. It is a moral issue. One of 10 Commandments says that “You shall not steal”. Man has the right to private property. The others must respect it. Man has no right to call on foreign thing that he does not belong. Man cannot illegally usurp something that is not his. Breaking copyright is equal to steeling. On Mount Sinai, God gave Moses 10 Commandments to be followed. They tell us how to live. Transgression of the commandments by a person is a committing sin. It is always derived from the specific needs of a particular type of subject matter and the environment against which protection is required. Protection solution of author's work in electronic (or digital) form in the online environment requires a specific approach and specific methods. In general it is true that 100% protection against abuse is not possible. Therefore, it is necessary to seek out ways of protecting copyright, which have a preventive effect and costs of their application are minimal.

Keywords: copyright protection, photography, digital form, steganography, Matlab

1. Introduction

You shall not steal. The neighbour property includes among material things, his reputation, public recognition, thoughts and trust that one gets from others. If someone appropriates some of its intellectual property, and robs the trust of others by disseminating information about his private ratios, exceeds this commandment even more serious than the injury to tangible property [Desať Božích prikázaní, accessed 02.05.2014, http://www.poznanie.sk/clanky/desat-bozich-prikazani.php].

In this sense we can indicate the copyright protection as one of the Ten Commandments. Problematic of copyright is paid much attention now. The Slovak Act defines who the author is, what the work is, and how the work and the author are protected. On the other hand, literature gives an example from practice, from which it is clear that in case of dispute, the burden of proof is almost always on the author [K. Babiaková, *Praktické rady ako chrániť autorské*

[*] E-mail: robert.halenar@ucm.com

práva, 6, accessed 20.03.2014, http://www.ephoto.sk/photopointy/photopointy-cz/vysocina/prakticke-rady-ako-chranit-autorske-prava/].

„Photographer drafted and passed a number of photos for the client, which, however, did not pay for ordered photos. Photographer applied to the court for payment of the amount owed, and over the course of the proceedings mentioned that some photos did his son, while the court did not submit slides of those pictures. The Court therefore decided in this proceeding as follows: photographer did not show in court proceedings that he is the author of all photos. Photographer was not eligible for proposal, and therefore the court rejected his proposal. Court decision that the burden of proof as a procedural responsibility of the participant for the outcome of the case means the party who has failed to adduce evidence necessary to support his statements carries a possible adverse effects such as a court decision, which will be based on the facts ascertained on the basis of other evidence transferred.“ [K. Babiaková, *Praktické rady ako chrániť autorské práva*, 6]

There are several ways to protect the rights of the author, but their implementation is largely dependent on the form of the work. What can be regarded as a work is described in the Act 618/2003 from collection laws, paragraph 7th: „Is subject to copyright and other literary work of art and scientific work, that which is the result of his own intellectual creation of the author, especially:

a) literary works and computer programs;
b) oral, submitted or otherwise made literary work, especially speeches and lectures;
c) theatrical works, especially the dramatic work as musical works, choreographic and pantomime work and other work created for publication;
d) musical work with text or without text;
e) audiovisual work, in particular cinematographic work;
f) painting, drawing, sketch, illustration, sculpture and other works of visual art;
g) photographic work;
h) architectural work, especially work of building architecture and urban planning, work of garden architecture and interior design and construction work;
i) works of applied art;
j) cartographic work in analogue or in another form.“ [Act 618/2003 collection laws, accessed 20.03.2014, http://www.vyvlastnenie.sk/predpisy/autorsky-zakon/]

2. Copyright protection

Photography is historically a strict selective medium. This medium, excepting the principle of reproduction, attributed to reality its reverse photogenic system, which we see as signs, artefacts and iconography [1].

If the photograph meets the conditions of its own intellectual creation of the author, is considered a photographic work. Author of such photographic work obtains a photo copyright which consists of personality and equity components. Creation of the copyright to the photography is automatic and it is no need for the registration, but it brings disadvantages as well, because without registering it is not clearly and easily demonstrable, who has the copyright. This condition is favouring the plagiarists. Author of the photography, to prove his rights, must use all means legal and technical, to make his authorship of the picture clear and demonstrable. One of the basic things that every author should do is to realize his basic right and photography mark with his name or a pseudonym. Marking the photo can be done by inserting the author's name directly on the photo or photo below, while the menu can be supplied with the character © (copyright) and date of the work creation. Indication of authorship can also be put under the photo, or next to it. To indicate the name of the author photo in electronic form, there are several technical methods, which should act as a preventative proof of authorship [K. Babiaková, *Praktické rady ako chrániť autorské práva*, 2-4].

2.1. Methods of protection

2.1.1. EXIF

Keep the original photos with precise and original EXIF (Exchangeable image file format – information set, which are attached at each image that the camera will shoot). EXIF can carry information about:

- the author,
- date and time of design or date of the last modification of the picture,
- data on the camera and lens (brand, exact model),
- exposure data (exposure time and aperture values and also other camera settings)

These are the basic information that a metadata file can contain. The advantage is that inside EXIF in some programs, you can write the name or other information about authorship. Some programs can even lock this information. The disadvantage, however, is that many programs allow you to change information in the EXIF, even those that are locked. Basically every software and digital lock is attacked by hackers [K. Babiaková, *Praktické rady ako chrániť autorské práva*, 4-5].

2.1.2. Authorization software

Buy a more expensive software (around hundreds of euros) called Authorization software, which connects to every photo an unique code and registers photos in the database [K. Babiaková, *Praktické rady ako chrániť autorské práva*, 5].

2.1.3. Watermark

To images that we want to potentially sell them or prove something, we can bet watermark. Watermark (semi-transparent or translucent character that allows picture to be viewed and to evaluate the content and quality, but given that overlaps the picture, does not allow its use for commercial purposes or otherwise unlawfully use). However, it can be removed by a variety of software. The more complex and clearer watermark is, the harder is to remove it [K. Babiaková, *Praktické rady ako chrániť autorské práva*, 5].

2.2. Protection on Internet

All of these methods have a number of disadvantages. They are generally known or readily removable protecting authorship marks, or are relatively expensive. Photographers who shoot on film have a simpler demonstration of authorship in that they have more negatives, which should demonstrate that the person who owns them is also the author of the photographs. In the case of digital photography, proving is complex, even more if the photo is stored on a publicly accessible digital repository or the Internet.

Computers connected into network are strong and may serve to share your experiences represented by photography with your friend, family or just with public. But in the other hand, it may be risky when you allow access to your photo gallery for public. Some authors [2] are saying: "…PC could become an instrument of manipulation, corruption and dehumanization, of alienation for the ontological man, the spiritual being, face of God saved by the universal sacrifice of the Cross of Christ". If the photos are placed online, author cannot be sure by any system of the protection of his rights. Scripts 'right-click' can circumvent direct source view, drop down images can be circumvented in the same way, the watermark can be removed (sometimes with difficulty). Even if the photo is inserted into the Flash object, one can easily create a screenshot [L. Kyrnin, *How to Protect Your Digital Photos from Being Copied*, accessed 20.03.2014, http://webdesign.about.com/od/graphics/a/aa102406.htm]. In this case the author must find an alternative method of protection. A suitable alternative is to locate information directly to the user's profile photo (not the notorious metadata).

3. Steganography methods

The following formula provides a very generic description of the pieces of the steganographic process:

$$\text{cover\_medium} + \text{hidden\_data} + \text{stego\_key} = \text{stego\_medium} \qquad (1)$$

In this context, the *cover_medium* is the file in which we will hide the *hidden_data*, which may also be encrypted using the *stego_key*. The resultant file is the *stego_medium* (which will, of course be the same type of file as the cover_medium). The cover_medium (and, thus, the stego_medium) are typically image or audio files. In this article, I will focus on image files and will, therefore, refer to the *cover_image* and *stego_image*.

Before discussing how information is hidden in an image file, it worth a fast review of how images are stored in the first place. An image file is merely a binary file containing a binary representation of the colour or light intensity of each picture element (pixel) comprising the image.

Images typically use either 8-bit or 24-bit colour. When using 8-bit colour, there is a definition of up to 256 colours forming a palette for this image, each colour denoted by an 8-bit value. A 24-bit colour scheme, as the term suggests, uses 24 bits per pixel and provides a much better set of colours. In this case, each pixel is represented by three bytes, each byte representing the intensity of the three primary colours red, green, and blue respectively (RGB). The Hypertext Markup Language (HTML) format for indicating colours in a Web page often uses a 24-bit format employing six hexadecimal digits, each pair representing the amount of red, blue, and green, respectively. The colour orange, for example, would be displayed with red set to 100% (decimal 255, hex FF), green set to 50% (decimal 127, hex 7F), and no blue (0), so we would use "#FF7F00" in the HTML code.

The size of an image file, then, is directly related to the number of pixels and the granularity of the colour definition. A typical 640x480 pixels image using a palette of 256 colours would require a file of about 307 KB in size (640 • 480 bytes), whereas a 1024x768 pixels high-resolution 24-bit colour image would result in a 2.36 MB file (1024 • 768 • 3 bytes).

To avoid sending files of this enormous size, a number of compression schemes have been developed over time, notably Bitmap (BMP), Graphic Interchange Format (GIF), and Joint Photographic Experts Group (JPEG) file types. However, not all are equally suited to steganography [G. Kessler, *Steganography: Hiding Data Within Data*, accessed 27.04.2014, http://www.garykessler.net/library/steganography.html].

4. Matlab environment

It is basically a simulation programming language integrated into the 4th generation integrated environment for scientific and technical calculations, modelling, simulation and data analysis. It allows working interactively, but also to build an application. It provides users with a relatively powerful graphics and computational tools, but also an extensive library of functions that are in scope usable in virtually all areas of human activity. Thanks to its designed architecture Matlab is for those who need to solve computationally intensive tasks without detailed examination of the mathematical nature of the problem. Custom MATLAB is much easier than Delphi and C and provides great potential for productivity and creativity. A major strength of Matlab is the fast computational core with optimized algorithms and a strong mathematical base. The Matlab implementation is on all key platforms – Windows, Linux and followers.

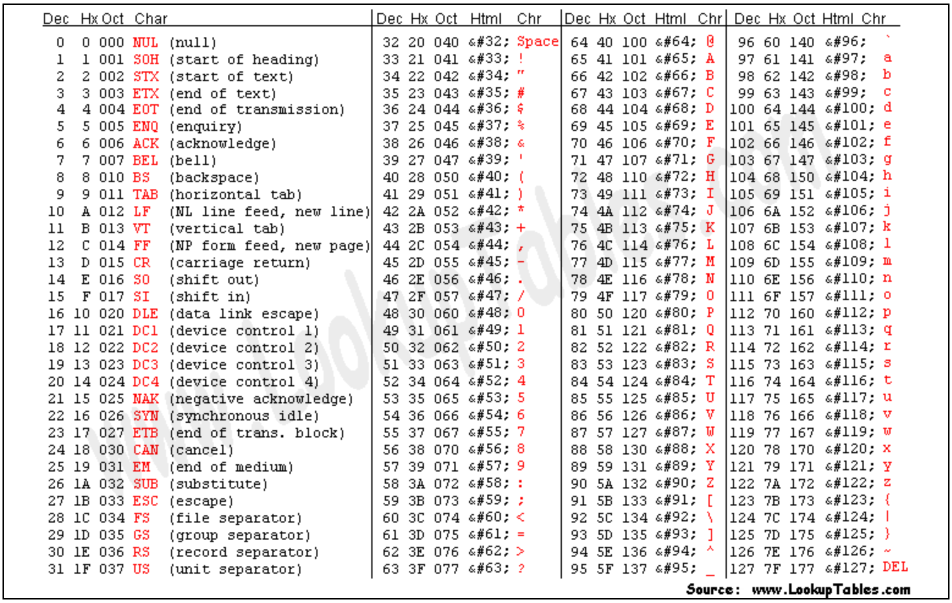
Matlab graphical options make it easy to display and present the obtained results. It is possible to render different types of charts - from two-dimensional, through histograms, and the three-dimensional. Graphical representation can be presented in multiple windows simultaneously, and may display multiple graphs in one window. The three-dimensional graphs can achieve a higher plasticity of image shading by determining the source of the incident light; animation three-dimensional graphs show contours and many other graphics capabilities. Most of these effects can be achieved by one or a few commands. Matlab graphics system used to create controlled graphical user interfaces. There is a tool for creating interactive user interfaces.

An important and perhaps most important feature of Matlab is its open architecture. Matlab is a complete programming language, which allows users to create (program) their own functions for direct use in own calculations (programs) [3].

4.1. Steganography in Matlab

Data insertion directly into the photo can post information (cipher) which is not directly visible but can help in proving copyright. For place of ciphers we use directly pixels, which picture consists of. It is therefore a process which is similar to a watermark, but it is not directly visible (even in many cases indistinguishable to the human eye).

The principle consists in the code - simple password consisting of several characters (first name, last name, or other identifier), which is translated using the ASCII table numbers in decimal, which are converted into binary form, see Figure 1.

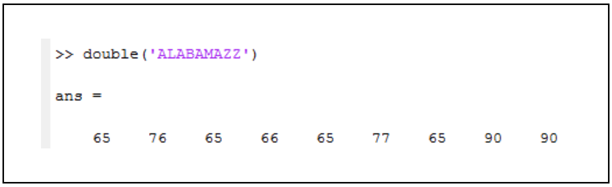


Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH	(start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX	(start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX	(end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT	(end of transmission)	36	24	044	&#36;	$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ	(enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK	(acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL	(bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS	(backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB	(horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT	(vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR	(carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO	(shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI	(shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE	(data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1	(device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2	(device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3	(device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4	(device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK	(negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN	(synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB	(end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN	(cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM	(end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB	(substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC	(escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS	(file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	|
29	1D	035	GS	(group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS	(record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US	(unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Figure 1. ASCII table [http://www.asciitable.com/, accessed 20.03.2014].

Let us Identifier ‘ALABAMAZZ’ that uses ASCII translate table as shown in Figure 2 (for automate translation was used MATLAB). Each character is expressed as a numerical representation. Then the decimal numbers are converted into binary form as shown in Figure 3.

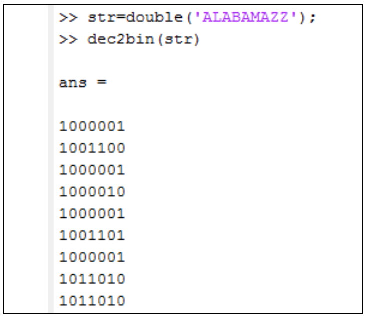


```
>> double('ALABAMAZZ')

ans =

    65    76    65    66    65    77    65    90    90
```

Figure 2. Decimal character codes identifier according to the ASCII table.



```
>> str=double('ALABAMAZZ');
>> dec2bin(str)

ans =

1000001
1001100
1000001
1000010
1000001
1001101
1000001
1011010
1011010
```

Figure 3. Decimal character codes identifier according to the ASCII table.

Via program routines in Matlab environment we can insert this information several ways. In this case we use LSB (least significant bit) to represent one bit described in Figure 3. An example of the script in Matlab in order to automate the whole process is presented in Scheme 1.

Every pixel is represented by its location (X and Y axis) and value of RGB (Red, Green and Blue basic colour). Each one rages from 0 to 255 (256 values – 2^8). This 8 – bit information give us LSB for placement of one bit of identifier. If identifier is 9 characters long and for every character is used a 7 – bit value (from ASCII table), we need 9 x 7 = 63 bits. In the RGB value of every pixel can be placed 3 bits. That’s why we need 21 pixels, if only LSB is used. Such a binary sequence is then encrypted into the individual pixels of the photo.

Figure 4 is a sample photo, which was coded. The left side is the original photo (no cipher), right side with encryption. Using conventional displaying equipment human eye can not recognize quality change, even when using detailed view (see Figure 5).

There are several ways to encode the identifier to photography. It is also possible to use areas with the same information, to place one bit – for example 10 x 10 pixels. This way we will be able extract information for identifier also if value of LSB in several pixels will be changed. We could use more massive penetration, when the last 2 or 3 bits are used with the cost of a reduced visibly quality of photography.

```
% Read phodography data into mattrix X
[X] = imread('c:\install\matlab\kvet.jpg');
mn=size(X);

% Generate binary code (9 characters password)
str=double('ALABAMAZZ');
k=0;
for i=1:9 %binary code in 9 characters, i.e. 9x7=63 bits, i.e. 63/3=21 pixels
  b=dec2bin(str(i));
  for j=1:7
    k=k+1;
    code(k)=b(j);
  end;
end;

% LSB value increased by „code bit“ in 21 pixels (63:3)
b=0;
  for i=1:mn(1)
    for j=1:mn(2)
      for k=1:3
        % Prepare RGB in mattrix X, place zero in LSB
        if mod(X(i,j,k),2) ~= 0
          X(i,j,k)=X(i,j,k)-1;
        end;

        % LSB value increased by „code bit“ in 21 pixels (63:3)
        if b<63
          b=b+1;
        else
          b=1;
        end;
        X(i,j,k)=X(i,j,k)+bin2dec(code(b));
      end;
    end;
  end;

% Write photography with code „ALABAMAZZ“ password imwrite(X,'c:\install\matlab\write\img1b.bmp');
```

Scheme 1. The sample of code in Matlab environment, which represents steganography routine.

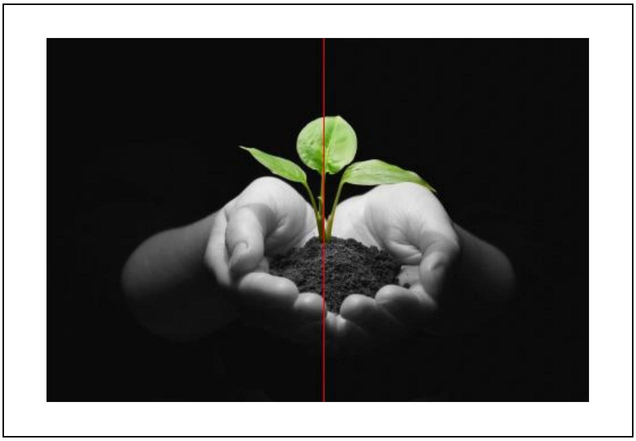
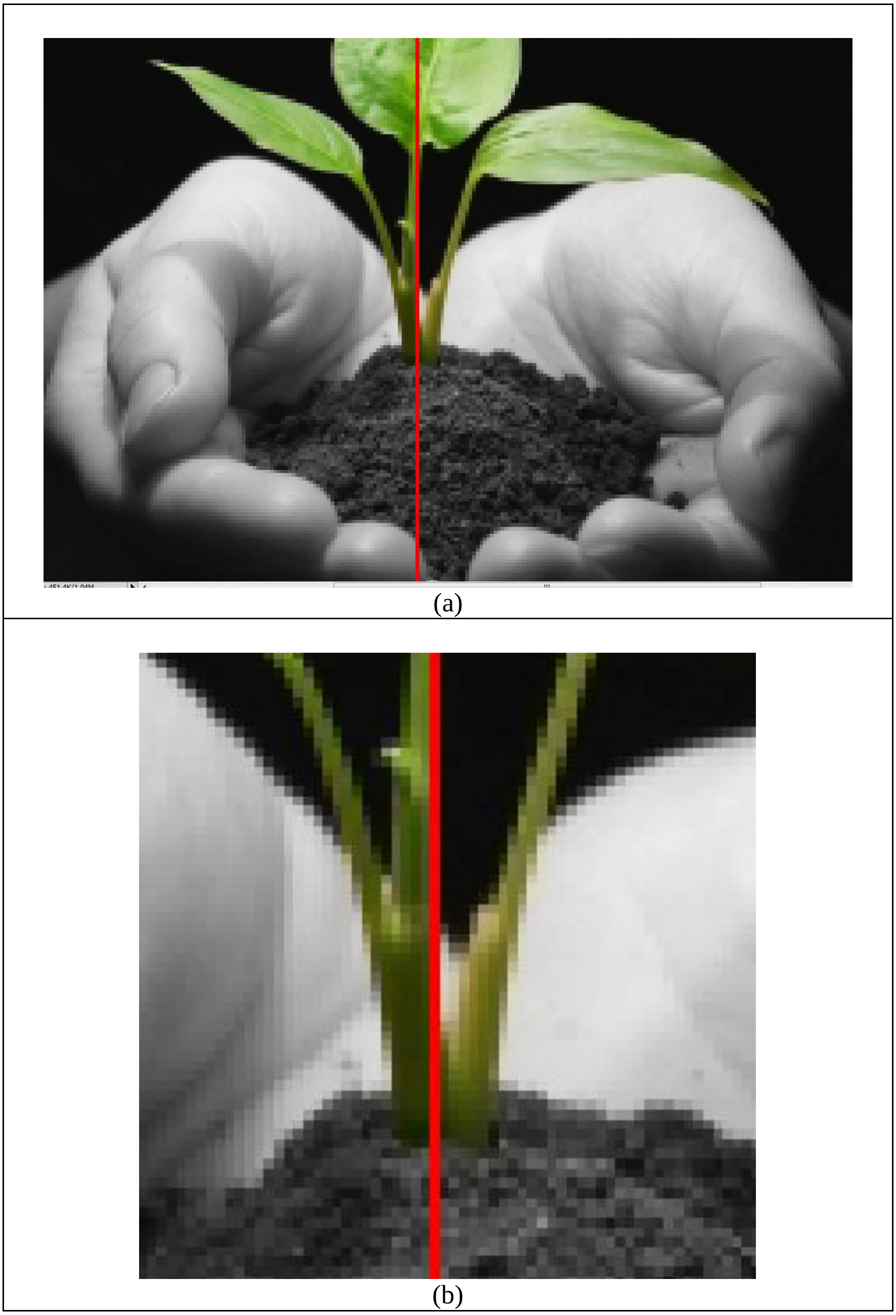


Figure 4. Sample photo with the application of binary code [http://www.garagebiz.ru/view/glavnaya_prichina_neudach_startapov_-_prezhdevremennoe_masshtabirovanie/startapi, accessed 20.03.2014].



(a)

(b)

Figure 5. Sample photo with the application binary code: (a) 2x zoom detail, (b) 4x zoom detail.

Then conducted data analysis of the extracted data may show that the identifier is located at the location of the bit structure, respectively that the data has been changed and there has been a fraud. Literature suggests several methods of analysis, but nowadays it is appropriate to focus on methods that allow to process large amounts of data [4].

Identifier 'ALABAMAZZ' was coded into photography, which does not visibly change its quality and is indistinguishable to the human eye, but can be measured and subsequently reconstructed. The whole code process is automated in Matlab environment, which is very suitable for mathematical operations [5]. The identifier can be added to the pictures showed in example with a resolution of 480x321 dots (pixels) more than 7300 times in a few seconds. Consequently, a part of the photography used by plagiarist will also include a sufficient sample for back remodelling identifier. Placement code system is therefore suitable for use in an online environment to prove the origin of the copy, when someone manipulates regardless the copyright.

5. Conclusions

Photography, is an expression of the author, his testimony. Using his work without his knowledge we cheat our neighbour. By ignoring the copyright law we are breaking one of the Ten Commandments.

Stealing is defined as "taking another person's property without his or her permission". However, there are many other forms of theft. The apostle Paul, when discussing God's commandments, sums up the entire law in the same way as our Lord Jesus did, with "Love your neighbour as yourself" (Mark 12.31, Romans 13.9). And, again like Jesus, he states that this is the fulfilment of the 'Law' (Matthew 22.39-40). So, we know from such instructions that 'Do not steal', as with all of the Ten Commandments, is about "loving one another" (John 13.34-35) [S.M. Houdmann, *Why is "You shall not steal" in the Ten Commandments?*, accessed 08.04.2014, http://www.gotquestions.org/you-shall-not-steal.html].

References

[1] J. Sedlák, Communication Today, **1(2)** (2013) 118-126.

[2] I. Rusu and G. Petraru, Eur. J. Sci. Theol., **1(1)** (2005) 3-9.

[3] R. Halenár, *MATLAB - Properties of dynamical systems: research of dynamic system properties*, LAP LAMBERT Academic Publishing, Saarbrücken, 2012, 62.

[4] A. Trnka, *Six Sigma Methodology with Fraud Detection*, Proc. of 9th WSEAS Interanational Conference on Data Networks, Communications, Computers, International Association of Engineers, Hong Kong, 2010, 162–165.

[5] R. Halenar, *Matlab Routines Used for Real Time ETL Method*, Proc. of the 4th International Conference on Mechanical and Electrical Technology, Trans Tech Publications, Durtnten-Zurich, 2012, 2125–2129.