ALLEGORICAL IMAGE

Table of contents

Description of the program

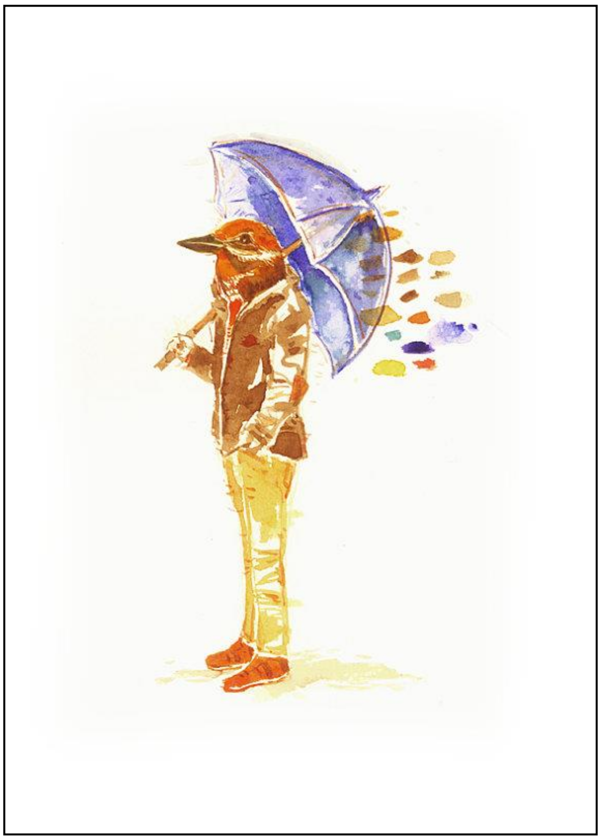
In the program I wrote, I imposed some restraints to optimize the result.

1. The hidden image is converted to a grayscale. Therefore, I receive one component of a pixel instead of three, as a result I can encode three times bigger image than if I encode a colored image.
2. A number of pixels of the hidden image cannot be larger than $\frac{3}{8} - 10$ of pixels of the basic image. This is due to the fact that there are three components of a pixel (hence the multiplication by 3) and that I use 8 bits to encode one color (hence the division by 8). 10 is a number of pixels reserved for encoding the hidden image's height and width. It is necessary for a proper decoding of the image.
3. Height and width of the hidden image is encoded on 10 pixels, which gives 30 bits, 15 for each value. This implies that the maximum dimension of the hidden image possible to be saved (without undesirable consequence in a work of the program) is 16384 x 16384 pixels.

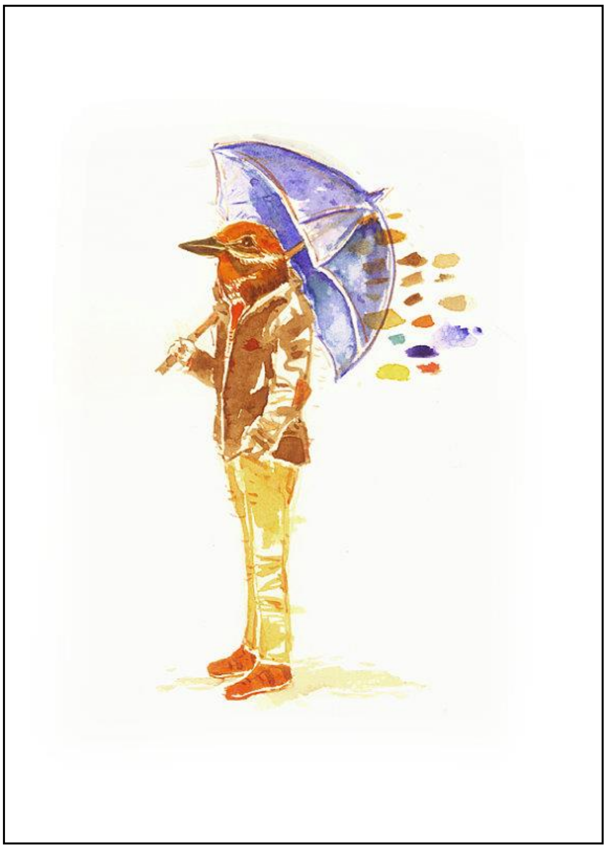
The pattern of the program:

1. In the beginning of the program it is necessary to select images destined to operation. Without it program will return an error.
2. The program calculates dimensions of images and checks whether condition included in the second point of previous list is fulfilled.
3. The image destined for hide is converted to a grayscale. Values got in this operation are spread into bits and written into an array.
4. The next step is changing bits of every component of a pixel in the basic image in R, G, B order until the array with spread values of the hidden image ends. If that array ends, next bits aren't changed. Last 10 bits are reserved for encoding height and width of hidden image.
5. A new image with encoded hidden message is saved.

Illustrative application of the program



Basic image



The same image after changing the bits

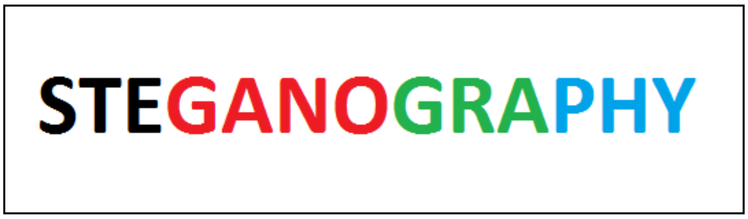


Image destined to hide

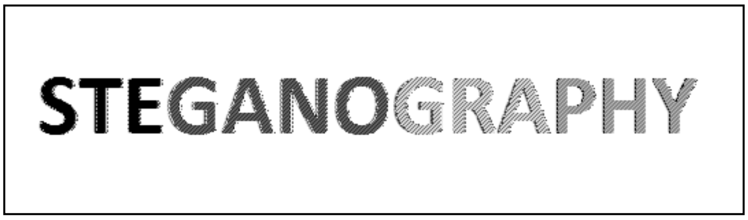
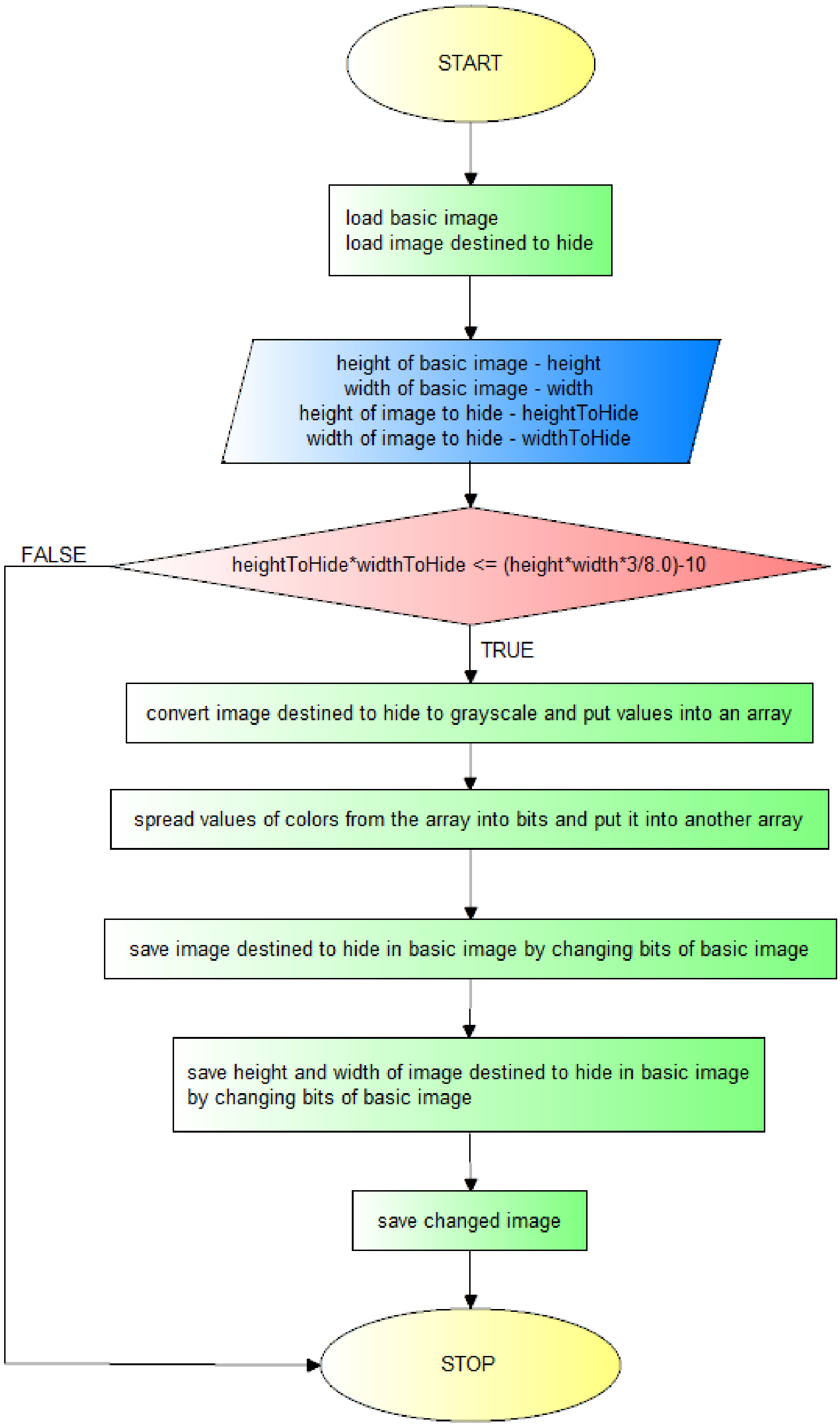


Image decoded from basic image with changed bits

Simplified flowchart of the program


START
load basic image
load image destined to hide
height of basic image - height
width of basic image - width
height of image to hide - heightToHide
width of image to hide - widthToHide
heightToHide*widthToHide <= (height*width*3/8.0)-10
FALSE
TRUE
convert image destined to hide to grayscale and put values into an array
spread values of colors from the array into bits and put it into another array
save image destined to hide in basic image by changing bits of basic image
save height and width of image destined to hide in basic image
by changing bits of basic image
save changed image
STOP

The flowchart was drawn with Diagram Designer program.