**PERSPEKTIV**

**An Image Camouflaging Tool using Thayer Counter Shading**

Prepared by

Amar B.P  
Karthik M Holikatti  
Chethan S

(7th Semester CSE)

**Contents**

1. What is an SRS? …………………………………………………………………. 1
2. Description ………………………………………………………………………. 1
3. Perspective ………………………………………………………………………. 2

4a. Functional Requirements ...........………………………………………................ 2

4b. Non-Functional Requirements ............................................................................... 2

5. User Characteristics …………………………………………………………......... 3

6. Constraints ………………………………………………………………………... 3

1. Assumptions and Dependencies …………………………………………............ 3
2. User Interface …………………………………………………………………… 4
3. Software Interface ……………………………………………………………..... 4

10. Hardware Interface ……………………………………………………………..... 4

11. Communications Interface …………………………………………………......... 4

**What is an SRS?**

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for [software](http://searchsoa.techtarget.com/definition/software) under development. The SRS fully describes what the software will do and how it will be expected to perform. A good SRS defines how an [application](http://searchsoftwarequality.techtarget.com/definition/application) will interact with system [hardware](http://searchcio-midmarket.techtarget.com/definition/hardware), other programs and human users in a wide variety of real-world situations.

**Description**

We are attempting to make a software that will take a bitmap format picture and hide the main object of the image in the background of the image. The real image can be obtained using an encryption key which will be known, only by the sender and receiver.

The idea for this project is based on the paintings of Abbott Handerson Thayer. **Abbott Handerson Thayer** (August 12, 1849 – May 29, 1921) was an American artist, [naturalist](https://en.wikipedia.org/wiki/Naturalist) and teacher. As a [painter](https://en.wikipedia.org/wiki/Painting) of [portraits](https://en.wikipedia.org/wiki/Portrait), figures, animals and [landscapes](https://en.wikipedia.org/wiki/Landscape_art), he enjoyed a certain prominence during his lifetime, and his paintings are represented in the major American art collections. He is perhaps best known for his 'angel' paintings, some of which use his children as models.

We are using counter shading technique to match the intensities of the pixels in the image. An image will be disintegrated into a large number of pixels. Each pixel has 4 bytes containing the CMYK values (Cyan, Magenta, Yellow and Key). We are interested in the value of K, for all the pixels. We will run an iteration operation to find the Key values of all the pixels, find the average value and then change the Key values of all the pixels to this average value. By doing so, the intensities of all the pixels will be the same and hence the object of the image will be hidden in the image itself.

This software will be developed using C programming and a minimalistic User Interface.

**Perspective**

From the user’s perspective, this software will be an image encryption software or a camouflage software. It will have a minimalistic user interface. This project has mainly been designed for communication of images between defence personnel. But it can also be used as an image encryption software. The user will just have to load a bitmap image to the software and he will get a camouflaged or encrypted image.

From the programmer’s perspective, this software will input a bitmap image. Counter shading will be used to carry out this main functionality. It will be split into its constituent pixels. A histogram of the pixels against their Key values will be plotted. The Key values of the pixels are stored in an array. An iteration operation will be run on this array to find the average Key value of all the pixels. The Key values of all the pixels will be changed to this value.

**Functional Requirements**

The software will have the following functional requirements:

1. Basic UI: The software will have a basic UI for interaction with the user.
2. Input image: The software will provide a dialog box to choose an image from the file system.
3. Parameter Satisfaction: The software will check whether the image is compatible with all its required parameters, to perform the encryption/camouflaging.
4. Encryption/Camouflaging: Once the parameters are satisfied, the image will undergo processing through the software’s algorithm. The camouflaged image will then appear on the screen as the output.
5. Save Encrypted Image: The user will be able to save the output image to their file system.

**Non-Functional Requirements**

1. Image format Conversion: The image uploaded by the user will most likely be in a jpg, jpeg, png format. This has to be converted to a bmp format.
2. Image Colour Scale Conversion: The uploaded image will most likely be in an RGB scale. This has to be converted to a CMYK scale.
3. Performance: The performance of the software has to be tested thoroughly using different types of images.
4. Capacity: The software must be able to handle images of all sizes. It must not crash if an image with High Resolution is used.
5. Usability: The software has a basic UI. This makes it very easy to use for any type of user.

**User Characteristics**

As discussed earlier, this will be an encryption software. It is primarily developed for communication of images between defence personnel. However, it can also be used by any person as an image encryption or camouflage software.

The main functionality is to hide the content of the image in the image itself. This is the main feature of the software. However, it will also have different levels of encryption depending on the user’s requirement. This additional feature makes it safer as it becomes harder to decrypt.

**Constraints**

1. This software can only be used to encrypt bitmap images. It does not work on layered images i.e images of another format.
2. Another constraint is that it needs a specific type of image. Not every image can be encrypted.

**Assumptions and Dependencies**

It is assumed that the user has a database of suitable images that can be camouflaged using the software. And that the images are in bitmap format. Another assumption is that the user knows how to interact with the interface.   
The user must have the appropriate software and hardware requirement to run the software.

**User Interface**

The user interface is very minimalistic with a simple dialog box containing a “load/browse” button to load the image from the file system. And it has a “submit” button to submit the selected image as the input to the software.

**Hardware Interface**

The software can be run on any PC which is capable of running a C program and supports bitmap image format.

**Software Interface**

The software interface is a dialog box with a “Load/Browse” button to load the image from the file system and a “submit” button to submit the selected image.

**Communications Interface**

The user interacts using the keyboard and the mouse.