**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
4. Product Functions ……………………………………………………………… 2
5. User Characteristics …………………………………………………………… 3
6. Constraints ………………………………………………………………………… 3
7. Assumptions and Dependencies ………………………………………… 3
8. User Interface ……………………………………………………………………. 4
9. Software Interface …………………………………………………………….. 4

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

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

**Description**

We are attempting to make a software that will take a image 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 RGBA values (Red, Green, Blue and Alpha). We are interested in the value of A, for all the pixels. We will run an iteration operation to find the Alpha values of all the pixels, find the average value and then change the Alpha 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 Java programming language 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 Alpha values will be plotted. The Alpha values of the pixels are stored in an array. An iteration operation will be run on this array to find the average Alpha value of all the pixels. The Alpha values of all the pixels will be changed to this value.

**Product Functions**

The software will have the following functions:

1. Software Loading
2. Starting the software

2a. About the Software

2b. Basic Requirements to use the software

2c. How to use the software?

1. Drag and drop box to drag and drop images
2. Dialog box to browse for the image in the file system
3. Set the encryption key.
4. Run the encryption.
5. Display encrypted image
6. Save the encrypted image in the file system.

**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 can 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**

* This software can take a lot of time to execute on an older OS.
* Even in the new version of a Windows platform, it could take up to 3 minutes.
* Another constraint is that it needs a specific type of image. Not every image can be encrypted.
* There must be a proper background in which the object can be hidden. And the object must not dominate the entire image.

**Assumptions and Dependencies**

* It is assumed that the user has a database of suitable images that can be camouflaged using the software.
* The images must suit the aesthetic requirements of the software. 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 Java 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.