Digital Image Watermarking

ECE 533 Image Processing, University of Wisconsin-Madison

Matthew Elliott and Brian Schuette

December 21, 2006

Introduction

Watermarking is a technique used to hide data or identifying information within digital multimedia. Our discussion will focus primarily on the watermarking of digital images, though digital video, audio, and documents are also routinely watermarked. Digital watermarking is becoming popular, especially for adding undetectable identifying marks, such as author or copyright information. Because of this use, watermarking techniques are often evaluated based on their invisibility, recoverability, and robustness. Our goal was to implement two different watermarking methods and evaluate their susceptibility to attack by various image processing techniques. Additionally, we wanted to create a GUI that would allow users unfamiliar with Matlab to add and extract watermarks, as well as evaluate their respective robustness based on a few morphological image attacks.

After learning about watermarking by bit-plane slicing in class, we were very interested to investigate the process by which one watermarks an image, as well as the degree to which the original image is changed by the watermarking process. To help us learn how images can be watermarked, we decided to implement two watermarking techniques, watermarking by bit-plane slicing and watermarking using the Cox method. It was extremely difficult to decide which watermarking methods to implement, because there are a multitude of different methods by

which to watermark an image. The Cox method and the bit-plane method allowed us to take two very different approaches to watermarking. We got to work in both the special and frequency domain, as well as having different goals for each method. Our bit-plane slicing approach is designed to work primarily as a fragile watermark. A fragile watermark shows the degree to which changes are made to an image. The Cox method, on the other hand, is designed to be robust. It works in the frequency domain, allowing it to resist many common attacks to the image.

In implementing these methods, we had to learn and create the processes to add a watermark and extract a watermark from digital images. To evaluate the degree to which watermarking affects the original image, the GUI was designed to display image difference graphically as well as numerically in a relative error format. This helps the user evaluate the invisibility of the watermark, as they can compare the changes watermarking makes to the original image. When the user extracts a watermark from an image, the difference between the watermarks is also shown both graphically and numerically. This will help the user decide if a watermark can be consistently recovered with the given method.

Approach

In applying watermarks, our focus was on invisibility, recoverability, and robustness. All of these are intricately linked. The less the image is affected, the easier it is to remove the watermark; recoverability is heavily reliant on robustness, for the watermark must still be present even after morphological attacks. Attacks may be accidental or intentional, but all images that

are digitally watermarked may be subject to attack. Most attacks are attempts to alter the image in order to destroy the watermark while preserving the image. Since watermarks may be hidden copyrights, this is extremely undesirable.

In order to address the issue of robustness, we decided to allow the user to use seven different morphological attacks to see how the extracted watermark is affected. The morphological attacks that are provided in the GUI are image scaling and cropping, as well as Gaussian low-pass (blur) filtering, unsharp contrast-enhancement filtering, averaging filtering, and circular averaging filtering. These attacks can be used to alter a watermarked image. The watermark can then be extracted and compared to the original watermark, allowing the user to evaluate the method's performance with respect to alterations. This allows the user to consider robustness in terms of recoverability, and how each of the methods stand up to various changes in the image.

To address the issue of invisibility, the GUI allows users to compare images before and after watermarking. It displays the difference visually and numerically. This same system is used to allow users to compare images before and after morphological attacks, allowing a wide spectrum of uses. With time, this GUI would aid in systematically identifying the strengths and weaknesses of various methods, allowing one to prepare counterattacks against the widest array of attacks possible.

In creating the project, we first began by implementing bit-plane slicing watermarking. Matlab functions to insert and extract a watermark were created. These functions work with the original image, as well as a binary watermark, which is inserted into the least-significant bit-plane as shown in Figure 1. The GUI and assisting functions were designed to accept color, grayscale, and binary images, which will be converted as needed when used as the original