

プログラミングコンテスト課題

電子透かしが埋め込まれた画像から乱数のシードを求める.

IT ハウス 盛 拓生

2013 年 10 月 4 日

1 はじめに

ネットワークを介したデジタルコンテンツの流通が一般化するにつれ、デジタルコンテンツの著作権保護が課題となっている。本稿ではプログラミングコンテストの課題として、デジタルコンテンツの著作権保護を保護する一手法である電子透かしをとりあげる。

2 電子透かし

電子透かしとは、動画像、静止画像、音声などのデジタルコンテンツに対して本来のコンテンツとは異なる何らかの情報を人間には知覚が困難となるように埋め込む手法を指す。

ここで埋め込まれた情報は電子透かしあるいは透かし情報等と呼ばれる。デジタルコンテンツが不正にコピーされた場合には、この透かし情報を抽出し、抽出された情報を利用してそのコンテンツの著作権者を特定したり、不正にコピーした人を特定したりすることに利用される。

電子透かしの対象となるデジタルコンテンツは多種多様なものがあり得るが、本稿では、静止画像を対象とし、その画像の画素値に対して透かし情報として乱数系列を埋め込む方法について述べる。また、プログラミング言語としては Java を利用することを前提とする。

また、透かし埋め込みに使用した Java 言語及び埋め込み対象画像、透かし埋め込み後の画像、使用する乱数系列の諸元を以下に記す。

- 埋め込み対象画像、透かし埋め込み後の画像のフォーマット: BMP
- 画像の幅: w ピクセル
- 画像の高さ: h ピクセル
- 画素中の埋め込み対象のビット: `java.awt.image.BufferedImage getRGB(x, y)` が返す `int` 型の数値の最下位ビット, つまり Least Significant Bit(LSB).
- 乱数生成器: `java.util.Random` クラス
- 乱数系列: `java.util.Random` クラスの `nextInt(int)` メソッドに引数 2 を渡して得られる乱数 $r \in \{0, 1\}^{w \cdot h}$

- 乱数のシードの生成方法: 秘密, ただしシードは非負の int 型の数値である.

2.1 透かし埋め込みアルゴリズム

1. java.imageio.ImageIO.read(File) メソッドにより透かし埋め込み画像を取得.
2. 非負の整数値を取る int 型のシードにより java.util.Random クラスのオブジェクトを初期化.
3. $0 \leq i < h$ について以下の処理を繰り返す.
 - (a) $0 \leq j < w$ について以下の処理を繰り返す.
 - (i) java.util.Random クラスのオブジェクトにより, nextInt(2) により $\{0,1\}$ の乱数を生成.
 - (ii) java.imageio.Image.read(File) により取得した java.image.BufferedImage クラスのオブジェクトより getRGB(j, i) により座標 (j, i) の画素値を int 型の数値として得る.
 - (iii) 3 (a) ii で得た int 型の数値の LSB を 3 (a) i で生成した乱数 ($\{0,1\}$) で置き換える.
 - (iv) 3 (a) iii で置き換えることにより得られた整数値で, 出力用の画像の (j, i) 座標の画素値を得る.
4. 全ての画素値の LSB を上記操作で置き換えて得られる画像を出力する.

3 課題の構成

課題となる電子透かし埋め込み画像には, Standard Image Data Base(SIDBA)[1] の標準画像より 10 の画像を利用する. これらの画像群に対して, 適当な方法により, 非負の値を取る int 型の数値である 10 のシードを生成し, シードから生成された乱数系列を電子透かしとして画像に埋め込む.

課題は, シードから生成された乱数系列を 2.1 節のアルゴリズムにより **電子透かしとして埋め込んだ画像が与えられたときに, その画像から乱数系列生成に利用されたシードを求める** ことである.

3.1 仕様

- パッケージ: pcon2013.extract
 - ソースコード
 - * ExtractingWatermarks.java
 - * FindBMP.java
 - 透かし埋め込み済み画像: pcon2013.watermarked 以下
 - * Aerial.bmp
 - * Airplane.bmp
 - * Balloon.bmp
 - * Earth.bmp
 - * Girl.bmp
 - * Lenna.bmp
 - * Mandrill.bmp
 - * Parrots.bmp
 - * Pepper.bmp

- * Sailboat.bmp
- 利用した乱数のシードのリスト
- * used_seeds.txt

3.1.1 ExtractingWatermarks.java

3.1 節で示したパッケージ `pcon2013.extract` を利用できるものとする。その前提で、次の仕様を満たすメソッドを作る。

入力は個々の透かし埋め込み済み画像を格納した `java.image.BufferedImage` オブジェクトであり、出力は入力画像に対して埋め込まれた電子透かしの乱数系列の生成に使用された乱数のシード (非負の `int` 型) とする。

```
pcon2013.extract.ExtractingWatermarks.findSeed(BufferedImage)
public static int findSeed(BufferedImage)
```

実行用の `main` メソッドは `ExtractingWatermarks.java` の `main` メソッドが利用できる。

3.2 評価

`ExtractingWatermarks.java` の `main` メソッド内で、3.1 内で示した SIDBA の 10 個の標準画像から乱数のシードを求める処理を 10 セット実行した場合の平均時間により評価する。

エクリプス上で実行可能であれば、最適化オプションの利用は自由とする。

- 評価基準
 1. 全ての画像について正しいシードが得られること。
 2. 全画像に対するシード探索処理を 10 セット実行したときの平均の実行時間が最短であること。

参考文献

- [1] 神奈川工科大学 情報学部 情報工学科 ESS 研究室,
”http://www.ess.ic.kanagawa-it.ac.jp/app_images-j.html,” (2013/10/02 アクセス)

Eclipse における開発環境構築方法

盛 拓生

2013 年 10 月 4 日

1. Pcon2013Extract.zip を適当なフォルダ上で解凍して Pcon2013Extract フォルダを得る.
2. Eclipse を起動
3. Eclipse 上で新規 Java プロジェクト, Pcon2013 を作成
4. 作成したプロジェクト Pcon2013 上を右クリック, インポートをクリック
5. インポート・ソースの選択からファイル・システムを選択
6. "次のディレクトリから"に対して, 参照をクリック, 1 で得た Pcon2013Extract フォルダを選択
 - (a) .settings, src, .classpath, .project にチェックを入れる
 - (b) 宛先フォルダは 3 で作成した Pcon2013 を指定, OK をクリック
 - (c) フォルダの上書きについて訪ねられたら, すべてはい, を選択
7. 以上で開発環境の導入は完了.
8. パッケージ, pcon2013.extract.ExtractingWatermark.java を Java アプリケーションとして実行する.
9. 実行後, 各自のホームフォルダ以下に, Pcon2013 及び Pcon2013¥watermarked が生成されることを確認する.
10. パッケージ pcon2013.extract.ExtractingWatermarks.java 内の

```
public static int findSeed(BufferedImage)
```

を編集して完成させる.