北京邮电大学计算机学院

实 验 报 告

课程名称： 信息隐藏与数字水印

实验名称： 隐写分析

实验完成人（姓名、学号）： 2018140757

指导教师：

日 期： 2018 年 12 月

# 一、实验目的

掌握RS隐写分析算法原理，仿真RS隐写分析算法并分析该算法性能。

# 二、实验内容

综合型实验：理解算法原理，综合运用图像信号处理和程序设计知识，进行算法仿真，分析算法性能。

# 实验环境

(1) Windows 8操作系统

(2) MATLAB R2015b

(3) 图像文件

# 四、原理

RS是一种针对LSB的隐写分析算法。其分析的依据是：不同类型的翻转操作，对自然图像和隐写图像的影响不同。这种影响通过空间相关性表现。RS算法给出如下函数表示图像块的空间相关性：



值越小，说明图像相邻像素之间的起伏越小。不同类型的翻转操作对空间相关性值的影响不同。RS算法考查三种翻转，分别是：零翻转（），正翻转（）和负翻转（）。其中，零翻转即不变关系，不改变像素值，因此，。正负翻转关系描述如下：



即表示值对（2i，2i+1）之间的变换关系，表示值对（2i，2i-1）之间的变换关系。

结合空间相关性函数定义如下3种像素组：R, S, U：



其中表示一组像素，表示依次翻转中每个像素。根据上述规则，如果一组像素被翻转后，它的空间相关性函数值变大了，那么这组像素称为正常像素块（Regular block）；如果相关性函数值变小了，那么这组像素称为异常像素块（Singular block）；如果相关性函数值不变，那么这组像素称为不变像素块。RS算法就是通过分析正常块（Regular block）和异常块（Singular block）的比例关系，来估计嵌入信息的长度的。

已知翻转有三种类型，表示像素块中像素的翻转类型的矢量称为掩模。由掩模确定的翻转函数矢量表示为 ，其中 为中像素。RS分析具体步骤如下：

将待检测图像（设隐写率为，则翻转率为）分成若干大小相等的图像块，通常选择4个像素为一块。

对图像进行非负翻转。

根据掩模翻转图像块中的各个像素。通常选择掩模为M(0, 1, 1, 0)，即。比较翻转前后，空间相关性函数值，判断此图像小块属于正常块（函数值变大）还是异常块。

遍历整幅图像，按上述方法对所有小块进行非负翻转。统计正常块和异常块相较于所有小块的占比，记为和。

对图像进行非正翻转。

按照步骤2方式，对所有图像块进行非正翻转，选择掩模为M(0, -1, -1, 0)。统计正常块和异常块相较于所有小块的占比，记为和。

产生翻转率为翻转为的新图像。

翻转原待检测图像（翻转率为）所有像素最低比特位。因此，相较于自然图像，在隐写时已经被翻转的像素，在本次操作中，像素值还原为自然图像时取值；在隐写时未被翻转的像素，在本次操作中，像素值翻转。所以，本次操作后，相较于自然图像，新图像翻转率为。

对新图像进行非负翻转。

按照步骤2方式，对所有图像块进行非负翻转，选择掩模为M(0, 1, 1, 0)。统计正常块和异常块相较于所有小块的占比，记为和。

对图像进行非正翻转。

按照步骤2方式，对所有图像块进行非正翻转，选择掩模为M(0, -1, -1, 0)。统计正常块和异常块相较于所有小块的占比，记为和。

解方程，计算隐写率。

令，，，,以及，，可得：

解方程，选其绝对值较小的根，可得隐写率（注：选择的是绝对值较小的根，但代入时，使用的“根”而不是“根的绝对值”）。

# 五、要求

## 5.1 RS隐写分析算法仿真

部份参考代码说明如下。

1. 非正、非负翻转及正常、异常块统计函数

输入参数为图像像素矩阵A和像素总数bufsize，函数返回值为非负翻转后正常块块数R、异常块块数S、不变块块数U，非正翻转后正常块块数R1、异常块块数S1、不变块块数U1。

function [R,S,U,R1,S1,U1] = RSU(A,bufsize)

R = 0;

S = 0;

U = 0;

R1 = 0;

S1 = 0;

U1 = 0;

for n = 1: bufsize/4

for m = 1: 4

x(m) = A(4\*(n-1) + m);

end

fG = f(x);

G(1) = F0(x(1));

G(2) = F1(x(2));

G(3) = F1(x(3));

G(4) = F0(x(4));

FsG = f(G);

if FsG>fG

R = R + 1;

elseif FsG < fG

S = S + 1;

else

U = U + 1;

end

H(1) = F0(x(1));

H(2) = Fne(x(2));

H(3) = Fne(x(3));

H(4) = F0(x(4));

FsH = f(H);

if FsH>fG

R1 = R1 + 1;

elseif FsH < fG

S1 = S1 + 1;

else

U1 = U1 + 1;

end

end

1. 空间相关性计算函数，用于计算相邻像素平滑度。

函数输入参数为相邻像素值向量x，返回值为其对应空间相关性函数值。

function sum = f(x)

sum = 0;

for n = 1:3

sum = sum + abs(x(n+1)-x(n));

end

1. 不变翻转函数。

function y = F0(x)

y = x;

1. 正翻转函数。

function y = F1(x)

if mod(x,2) == 0

y = x + 1;

else

y = x - 1;

end

1. 负翻转函数。

function y = Fne(x)

if mod(x,2) == 0

y = x - 1;

else

y = x + 1;

end

1. 主流程

准备图像，隐写率从0，以10%的步长，增长到100%，因此共有11幅图像。对每一幅图像都进行下面处理，以估计隐写率。

（1）将图像划分为4像素一组的小块。

（2）计算，，和。

（3）将所有像素最低有效比特位按位取反，计算相应的，，和。

（4）根据方程（详见原理）估计的隐写率。注意，方程中使用的参数是正常块和异常块与图像总块数的比例。

令，，，，解如下方程：

选其绝对值较小的根，可得隐写率（注：选择的是绝对值较小的根，但代入时，使用的“根”而不是“根的绝对值”）。

（5）记录不同隐写率情况下，实际隐写率与估计隐写率之间的误差。

隐写率0%：



隐写率10%：



隐写率20%：



隐写率30%：



隐写率40%：



隐写率50%：



隐写率60%：



隐写率70%：



隐写率80%：



隐写率90%：



隐写率100%：



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **隐写率** | **实际隐写率** | **估计隐写率** | **误差** |
| 1 | 0% | 0.000000 | -0.011645 | 0.011645 |
| 2 | 10% | 0.099609 | 0.116069 | -0.016069 |
| 3 | 20% | 0.199219 | 0.218273 | -0.018273 |
| 4 | 30% | 0.300781 | 0.300478 | -0.000478 |
| 5 | 40% | 0.400391 | 0.401739 | -0.001739 |
| 6 | 50% | 0.500000 | 0.486085 | 0.013915 |
| 7 | 60% | 0.599609 | 0.552346 | 0.047654 |
| 8 | 70% | 0.699219 | 0.626767 | 0.073233 |
| 9 | 80% | 0.800781 | 0.755934 | 0.044066 |
| 10 | 90% | 0.900391 | 0.826283 | 0.073717 |
| 11 | 100% | 1.000000 | 0.990396 | 0.009604 |

## 5.1 思考题

2.研究分块大小对算法的影响

分别按照1\*4，2\*2，1\*9，3\*3, 1\*16，4\*4方式划分图像块，比较相同嵌入率（例如，0，30%，50%）情况下，实际隐写率与估计隐写率之间的误差。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **嵌入率** | **图像块划分** | **实际隐写率** | **估计隐写率** | **误差** |
| 0 | 1\*4 | 0.000000 | -0.011645 | 0.011645 |
| 2\*2 | 0.000000 | -0.060360 | 0.060360 |
| 1\*9 | 0.000000 | -0.014404 | 0.014404 |
| 3\*3 | 0.000000 | -0.036975 | 0.036975 |
| 1\*16 | 0.000000 | -0.037043 | 0.037043 |
| 4\*4 | 0.000000 | -0.060360 | 0.060360 |
| 30% | 1\*4 | 0.300781 | 0.297593 | 0.002407 |
| 2\*2 | 0.300781 | 0.315948 | -0.015948 |
| 1\*9 | 0.300781 | 0.271713 | 0.028287 |
| 3\*3 | 0.300781 | 0.266920 | 0.033080 |
| 1\*16 | 0.300781 | 0.313450 | -0.013450 |
| 4\*4 | 0.300781 | 0.257671 | 0.042329 |
| 50% | 1\*4 | 0.500000 | 0.510531 | -0.010531 |
| 2\*2 | 0.500000 | 0.532905 | -0.032905 |
| 1\*9 | 0.500000 | 0.486689 | 0.013311 |
| 3\*3 | 0.500000 | 0.417263 | 0.082737 |
| 1\*16 | 0.500000 | 0.454477 | 0.045523 |
| 4\*4 | 0.500000 | 0.522807 | -0.022807 |
| 100% | 1\*4 | 1.000000 | 0.905375 | 0.094625 |
| 2\*2 | 1.000000 | 0.982930 | 0.017070 |
| 1\*9 | 1.000000 | 0.791173 | 0.208827 |
| 3\*3 | 1.000000 | 0.804993 | 0.195007 |
| 1\*16 | 1.000000 | 1.003872 | -0.003872 |
| 4\*4 | 1.000000 | 1.025413 | -0.025413 |

3.研究像素排列方式对算法的影响。

选取分块大小为2\*2和3\*3，像素按三种方式排列，比较相同嵌入率（例如，0，30%，50%）情况下，实际隐写率与估计隐写率之间的误差。三种排列方式为：列优先、行优先和ZigZag方式。以3\*3小块为例，x[i, j], i, j=0,1,2为小块像素，那么

列优先方式，像素序列为：

x[0,0],x[1,0],x[2,0], x[0,1],x[1,1],x[2,1], x[0,2],x[1,2],x[2,2],

行优先方式，像素序列为：

x[0,0],x[0,1],x[0,2], x[1,0],x[1,1],x[1,2], x[2,0],x[2,1],x[2,2],

ZigZag方式，像素序列为：

x[0,0], x[0,1], x[1,0], x[2,0], x[1,1], x[0,2], x[1,2], x[2,1], x[2,2],

|  |  |  |  |
| --- | --- | --- | --- |
| 嵌入率 | 图像块划分 | 排列方式 | 实际隐写率与估计隐写率之间的误差 |
| 0% | 2\*2 | 行优先 | 0.010424 |
| 列优先 | 0.024083 |
| ZigZag | 0.010424 |
| 3\*3 | 行优先 | 0.036975 |
| 列优先 | 0.037342 |
| ZigZag | 0.002408 |
| 30% | 2\*2 | 行优先 | -0.006073 |
| 列优先 | -0.018774 |
| ZigZag | -0.047479 |
| 3\*3 | 行优先 | -0.037771 |
| 列优先 | 0.073273 |
| ZigZag | -0.017061 |
| 50% | 2\*2 | 行优先 | -0.018874 |
| 列优先 | 0.013853 |
| ZigZag | 0.010993 |
| 3\*3 | 行优先 | 0.100601 |
| 列优先 | 0.101119 |
| ZigZag | 0.053466 |

# 七、附录

实验相关的MATLAB程序及其他材料见压缩包2018140757\_王泽\_信息隐藏分析.zip。