Steganography & Steganalysis

INSTRUCTOR: JOHN ORTIZ
SLIDES BY: MICHAEL SHARP
SENIOR COMPUTER ENGINEER
STEGO@SATX.RR.COM

PIXEL VALUE DIFFERENCING

Overview

- Pixel Value Differencing
- 2x1 Greyscale
- 2x1 24-bit Color
- Perfect Square
- Steganalysis

Pixel Value Differencing

- Mathematically manipulate difference in adjacent pixels to hide data
- Hide more data in edge sections of cover image
  - Sharp edges may have four times the capacity of smooth areas in some implementations
- Limited capacity in smooth images, e.g. cartoons

2x1 Greyscale: Method

- Divide image into non-overlapping 2x1 blocks
- Determine the difference between pixels in block to embed
- Find the difference in the range table
  - Used to determine how many bits may be hidden
- Determine what modified difference must be to hide message bits
- Determine how much original pixels must be adjusted to accomplish this
  - Adjustments are split evenly between both pixels

2x1 Greyscale: Range Table

- Width of each entry is a power of 2
- Wider ranges apply to edges
- Same table must be used for embedding and extracting
- Ranges can be made more narrow to reduce detectability
- Samples use range table of { 8, 8, 16, 32, 64, 128 }

range	width	bits hidden
[0, 7]	8	3
[8, 15]	8	3
[16, 31]	16	4
[32, 63]	32	5
[64, 127]	64	6
[128, 255]	128	7

2x1 Greyscale: Walkthrough

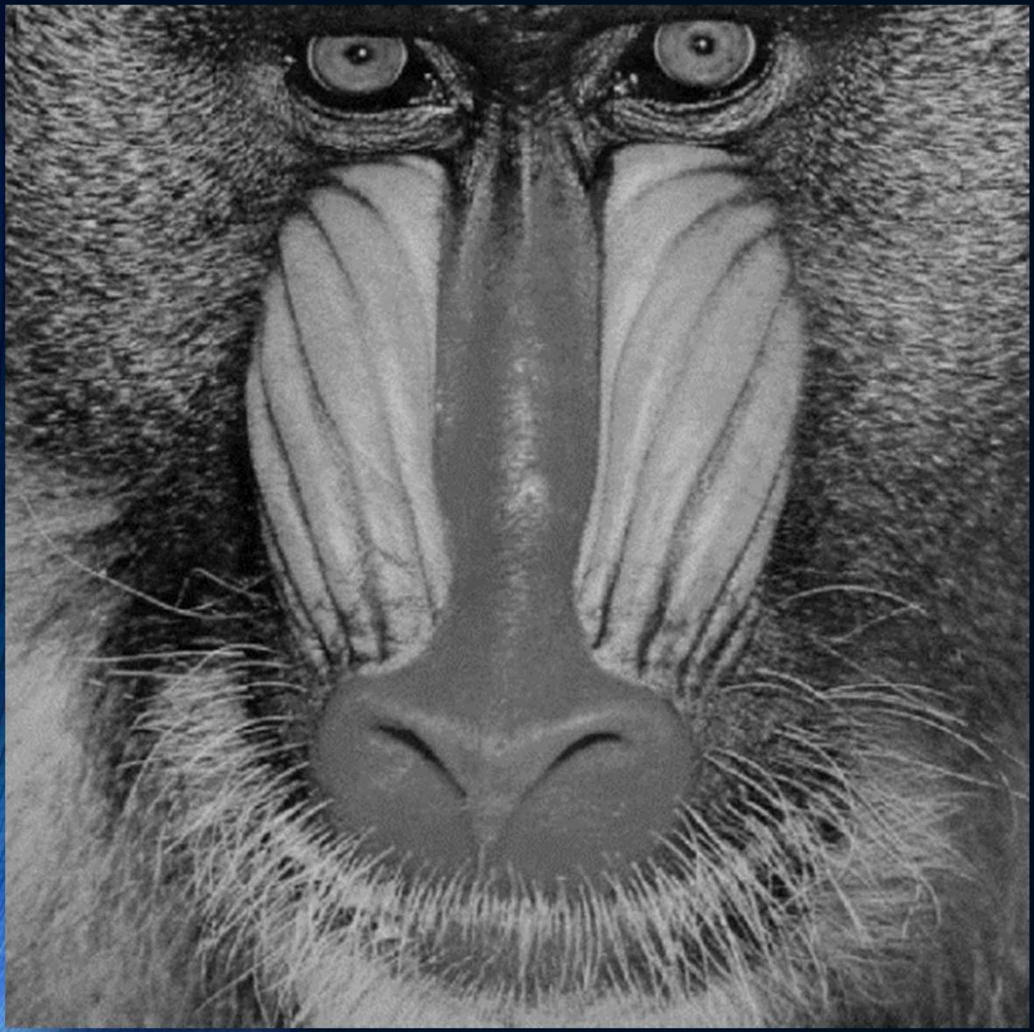
- Given g_i and g_{i+1} of (67, 109):
  - Difference: 109 – 67 = 42
  - Within range [32, 63]
  - Can hide 5 bits
- Given message bits of 10110 (decimal 22):
  - Target difference: 32 + 22 = 54
  - Difference needs to move by: m = 54 – 42 = 12
  - Original pixels are modified like so:
  - g'_i = g_i - [m/2]
    - = 67 – 6 = 61
  - g'_{i+1} = g_{i+1} + $\lfloor m/2 \rfloor$
    - = 109 + 6 = 115

2x1 Greyscale: Extraction

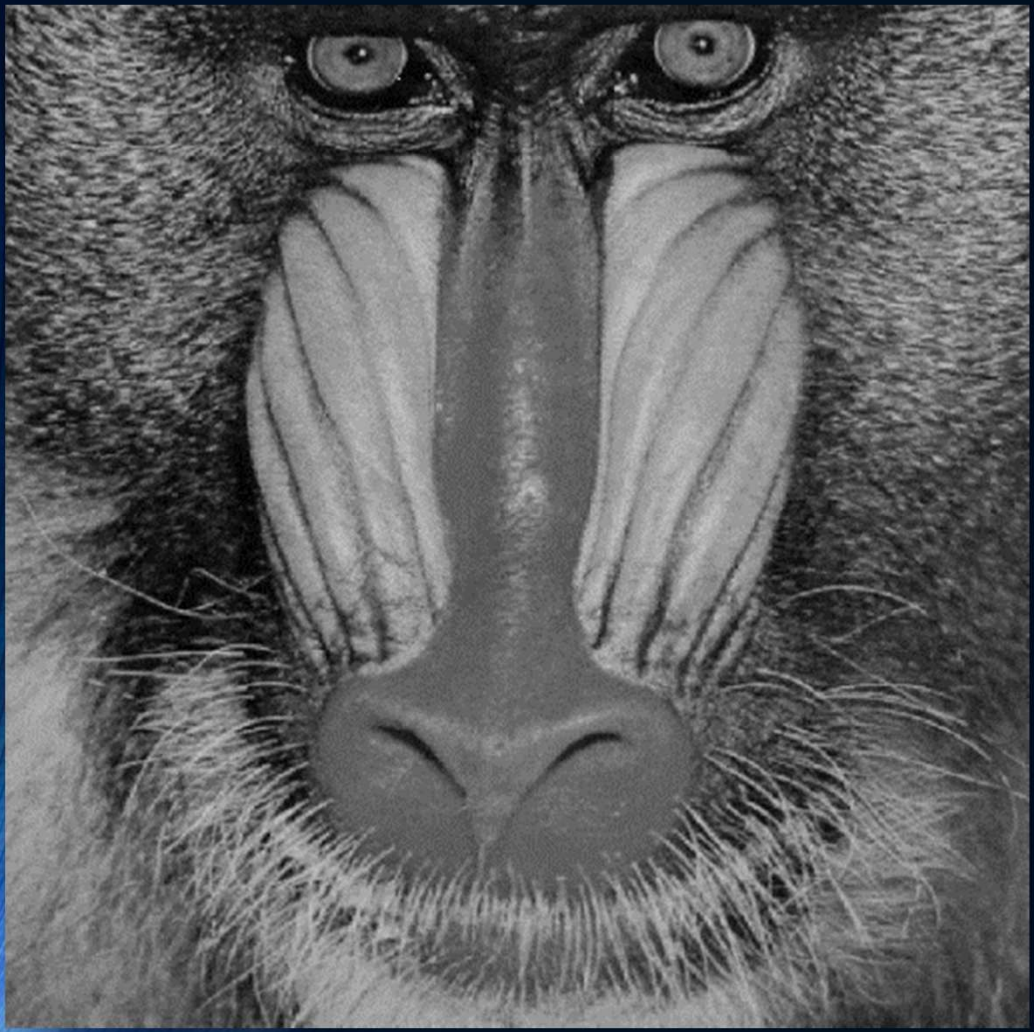
- Given values of (61, 115):
  - Difference: 115 – 61 = 54
  - Within range [32, 63]
  - Five bits were hidden
  - 54 – 32 = 22
  - 22 in binary is 10110, the hidden message bits

2x1 Greyscale: Falloff

- Modifying the values may cause them to fall out of the range [0, 255]
  - Becomes extremely noticeable changing between dark and light
- Ignore any blocks where this may occur during embedding and extracting
- Test by calculating modification when m is equal to the upper bound of the range minus pixel difference
  - m = 63 – 42 = 21
  - g_i = 67 – $\lceil 21/2 \rceil$ = 56
  - g_{i+1} = 109 + $\lfloor 21/2 \rfloor$ = 119
  - Data can be embedded here



512 x 512 Image
257 KB File Size
Hides 51 KB
Skipped 35 Blocks



512 x 512 Image
257 KB File Size
Hides 51 KB
Skipped 35 Blocks

24-bit Color: Method

- Divide the image into red, green, and blue matrices
- Hide in first red block, then first green, etc.
- Particular implementation used limits red and green ranges
  - Red <= 5 bits
  - Green <= 3 bits
  - Based on human visual system's ability to distinguish shades of red and green
  - Blocks are skipped if they exceed these maximums
- Implements adjustments to correct for falloff

24-bit Color: Falloff

- Does not skip blocks that would fall out of [0, 255] range
- Attempt to embed one less bit
  - Only attempted if MSB of message is a 1
  - Requires tracking which blocks embed one less bit
- If still falls out, force all of the movement into a single pixel
  - May cause perceptible distortion

$$
\begin{aligned}
(p_i', p_{i+1}') &= (p_i - m, p_{i+1}), && \text{if } p_{i+1} \ge p_i \text{ and } p_{i+1} \text{ crossing the upper range(i.e 255) ;} \\
(p_i', p_{i+1}') &= (p_i, p_{i+1} - m), && \text{if } p_{i+1} < p_i \text{ and } p_i \text{ crossing the upper range(i.e 255) ;} \\
(p_i', p_{i+1}') &= (p_i, p_{i+1} + m), && \text{if } p_{i+1} \ge p_i \text{ and } p_i \text{ crossing the lower range(i.e 0);} \\
(p_i', p_{i+1}') &= (p_i + m, p_{i+1}), && \text{if } p_{i+1} < p_i \text{ and } p_{i+1} \text{ crossing the lower range(i.e 0) .}
\end{aligned}
$$

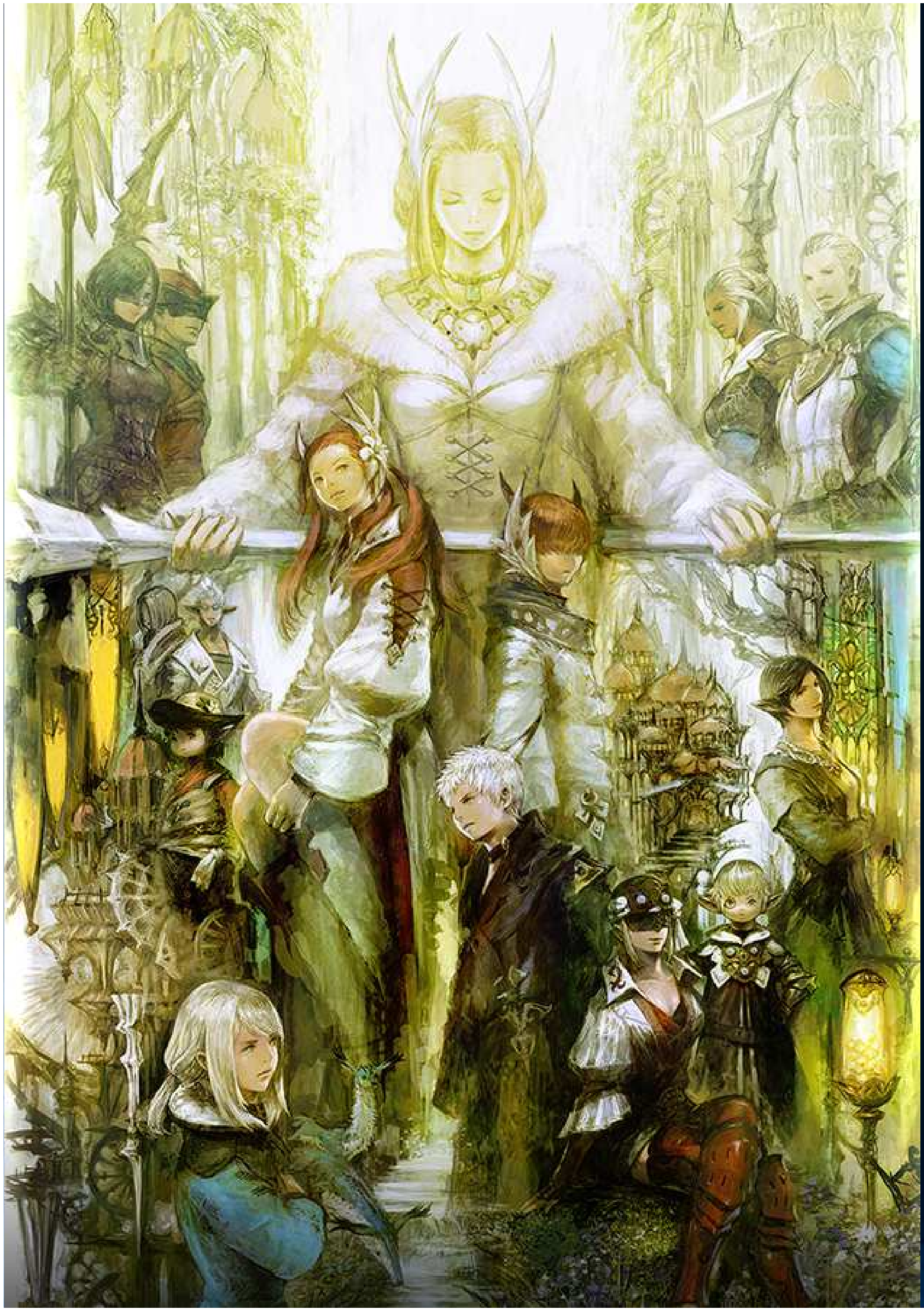
where $m = |d_i' - d_i|$.

640 x 930 Resolution
1.70 MB File Size
Hides 288 KB

640 x 930 Resolution
1.70 MB File Size
Hides 288 KB

632 x 900 Resolution
1.62 MB File Size
Hides 261 KB

632 x 900 Resolution
1.62 MB File Size
Hides 261 KB



632 x 900 Resolution
1.62 MB File Size
Hides 270 KB

632 x 900 Resolution
1.62 MB File Size
Hides 270 KB

632 x 900 Resolution
1.62 MB File Size
Hides 262 KB

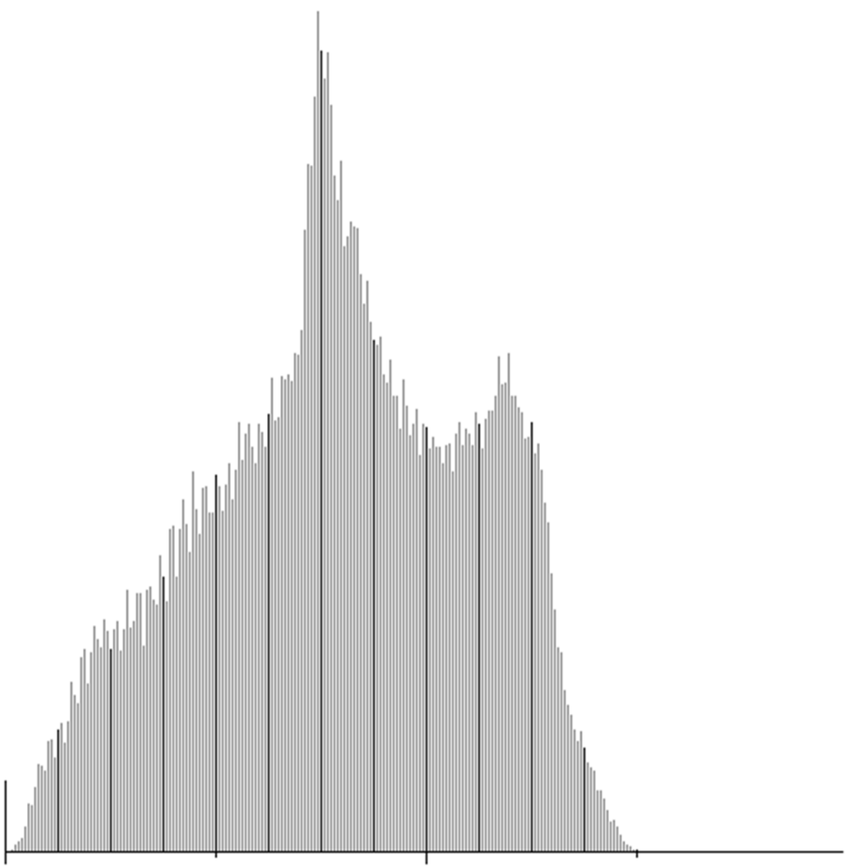
632 x 900 Resolution
1.62 MB File Size
Hides 262 KB

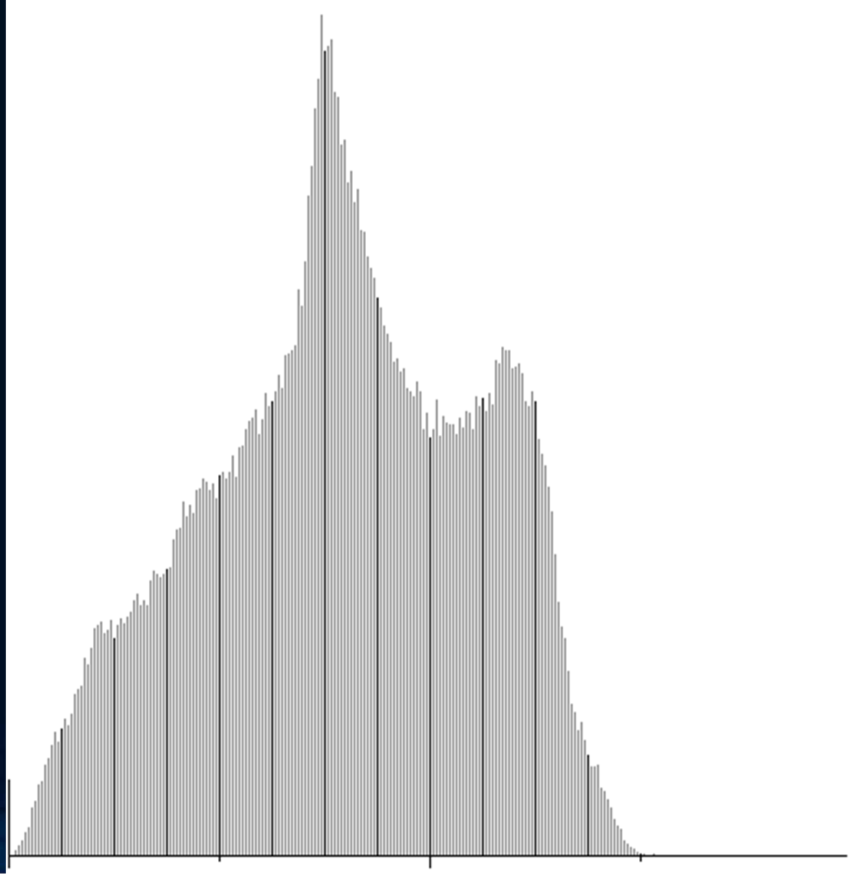
Perfect Squares: Advantages

- Reduces perceptibility even further
  - Especially true in smooth areas
- Gains capacity without sacrificing perception
  - Limited to subrange matches
- Better capacity and perceptibility than original PVD method
  - Tested against original PVD with range table { 2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, 64 }

PVD Steganalysis

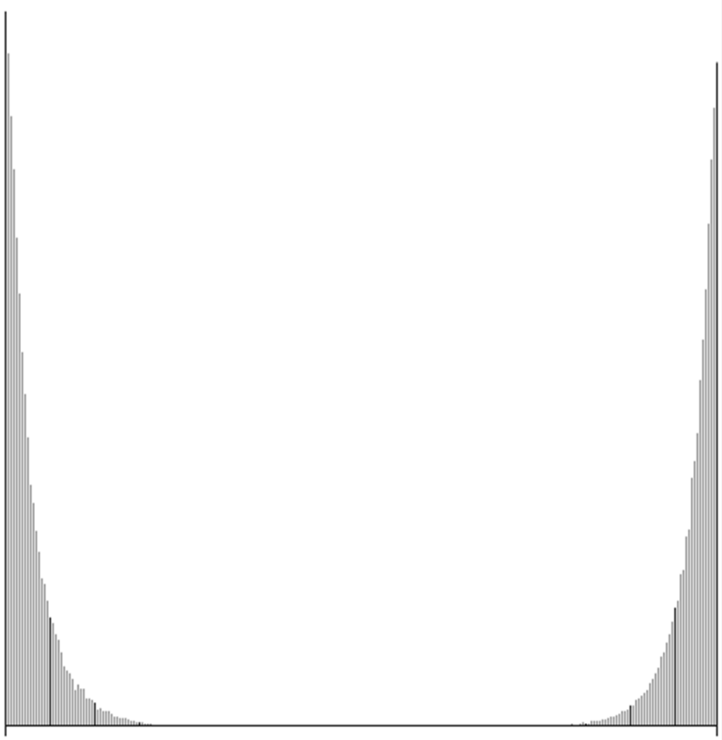
- Image histograms not very useful
- For color schemes, zooming in may reveal distortions

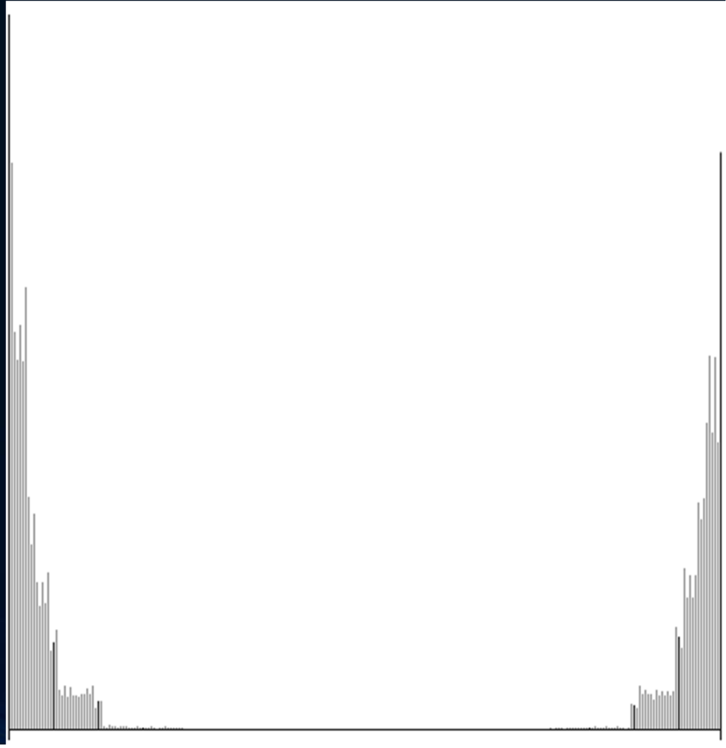




PVD Steganalysis

- Instead, calculate the pixel difference in every block
  - Skip any that would fall off [0, 255] range
- Histogram of these values will show evidence of embedding





Questions & Comments

References

- “Data Hiding In Gray-scale Images Using Pixel Value Differencing”, Samant, R. M. and Agrawal, S.
- “Colour Image Steganography Based on Pixel Value Differencing in Spatial Domain”, Mandal, J. K. and Das, Debashis
- “A Steganographic Method Based on Pixel-Value Differencing and the Perfect Square Number”, Tseng, Hsien-Wen and Leng, Hui-Shih
- “Steganalysis of Pixel-Value Differencing Steganographic Method”, Sabeti, Vajiheh et al.

Copyright Information