Region of Interest Signalling for Encrypted JPEG Images

Andreas Unterweger

Department of Computer Sciences
University of Salzburg

June 18, 2013

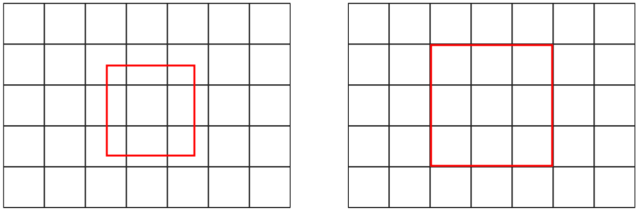
Overview

- Scope
  - New method to represent RoIs (location and size) compactly
  - New methods to embed this information into JPEG images
- Use case: Signal encrypted regions in JPEG images
  - Location and size of RoIs required for decryption
  - No separate signalling channel
- State of the art
  - Bitmap (1=encrypted/0=unencrypted)
  - Separate signalling channel
  - JPEG COM segment signalling

Practical considerations I

- Limitations
  - iMCU-granularity for coordinates (encryption use case)
  - 4:2:0 subsampling (surveillance cameras)
  - $\rightarrow$ $16 \cdot 16$ block size granularity

Note: Other block sizes and subsampling possible!



Practical considerations II

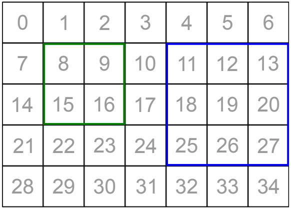
- Use of indices instead of X and Y coordinates
  - Index: Block number (starting with zero on the top-left)
  - Requires fewer bits than separate X and Y coordinates (proof in paper)
  - RoI size as index difference
  - → One RoI as tuple: $(\mathit{start}, \mathit{end} - \mathit{start} + 1)$ or $(\mathit{start}, \mathit{length})$

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34

Red RoI: $(9, 25 - 9 + 1) = (9, 17)$

Overview of the new coordinate encoding approach I

- Create list of index tuples based on location and size of RoIs
- Add "End of list" RoI $(0, 0)$



$\rightarrow$

Green RoI	$(8, 9)$
Blue RoI	$(11, 17)$
End of list RoI	$(0, 0)$

Overview of the new coordinate encoding approach II

- Keep first RoI tuple unchanged
- Encode tuple $n > 1$ relative to tuple $n - 1$ (differential coding)
- Keep "End of list" RoI tuple unchanged

Green RoI	$(8, 9)$		Green RoI	$(8, 9)$
Blue RoI	$(11, 17)$		Blue RoI **diff.**	$(3, 8)$
End of list RoI	$(0, 0)$	$\rightarrow$	End of list RoI	$(0, 0)$

- Differential values can be negative $\rightarrow$ signed values
- Differences are likely to be small $\rightarrow$ Exponential Golomb codes

Zeroth order Exponential Golomb codes

Value	ue(v) code word	se(v) code word
...	–	...
-4	–	0001001
-3	–	00111
-2	–	00101
-1	–	011
0	1	1
1	010	010
2	011	00100
3	00100	00110
4	00101	0001000
...	...	...

Based on H.264 k-th order Exponential Golomb Codes

Overview of the new coordinate encoding approach III

- Encode RoI tuples with one Exponential Golomb code word per value

Green RoI	$(8, 9)$		$000010000, 000010010$
Blue RoI diff.	$(3, 8)$		$00110, 000010000$
End of list RoI	$(0, 0)$	$\rightarrow$	$1, 1$

- Concatenate code words to one single bit string

$$000010000, 000010010|00110, 000010000|1, 1 \rightarrow$$
$$00001000000001001000110000010000011$$

Overview of the new coordinate encoding approach IV

- Optimization
  - Try all possible orderings of RoI tuples (does not affect signalling)
  - Use the one with the shortest encoded bit length
  - Beware of complexity (large number of RoIs)!

First green, then blue:	34 bits
First blue, then green:	34 bits

$\rightarrow$ Both ways equally short in this example

- Perform entropy coding on result (adaptive binary arithmetic coding)

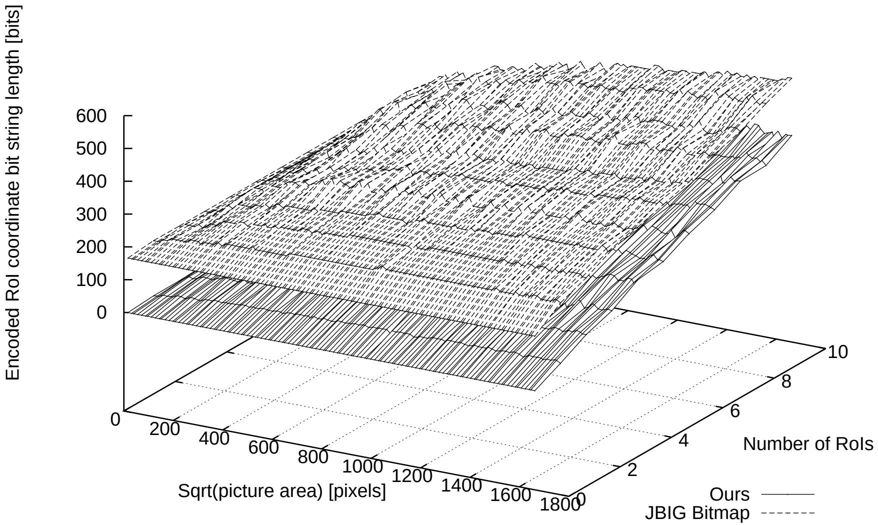
0000100000000100100011000001000011 $\rightarrow$
0010110000000000001010110010000

Results I: BEHAVEDATA surveillance data set

		Avg. encoded bit string length		
RoIs	**Pictures**	**Bitmap**	**JBIG* Bitmap**	**Ours**
0	46,382	1,200.00	176.00	*2.00*
1	1,444	1,200.00	212.34	*36.05*
2	3,449	1,200.00	231.46	*52.26*
3	2,820	1,200.00	238.52	*74.41*
4	1,877	1,200.00	245.66	*96.51*
5	10,822	1,200.00	260.68	*122.08*
6	814	1,200.00	267.50	*144.71*
7	3	1,200.00	266.67	*153.00*
8	2	1,200.00	256.00	*158.50*
Non-zero	21,234	1,200.00	248.64	*97.18*

* Modified (smaller) header

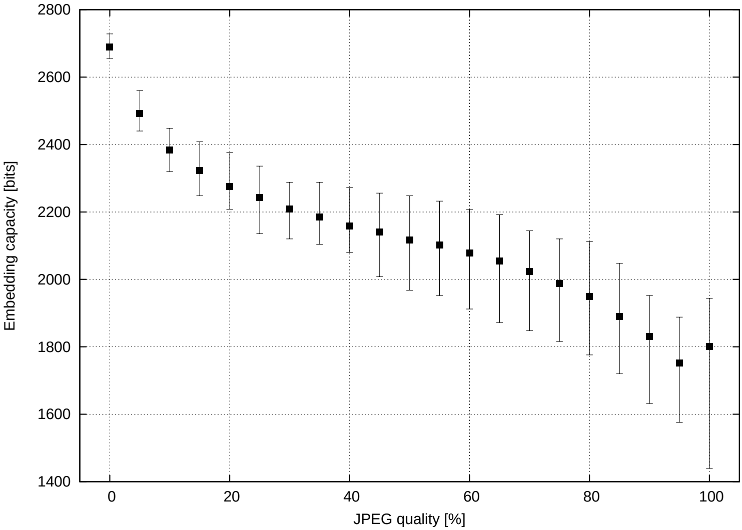
Results II: Artificial test data set



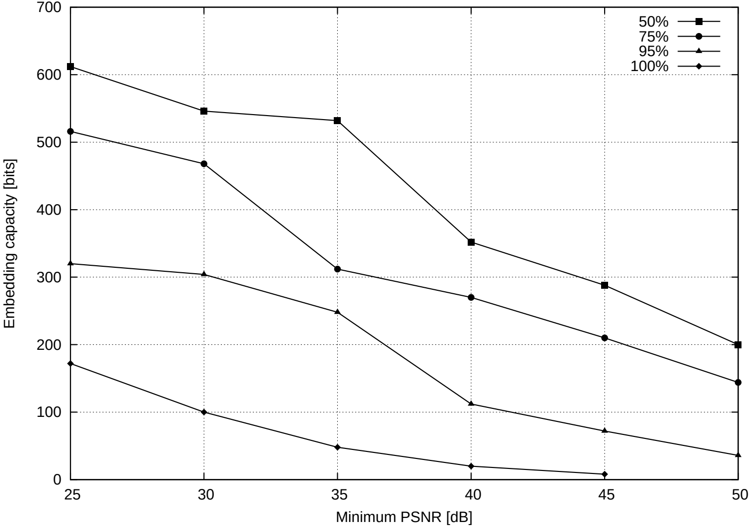
Overview of new RoI signalling approaches for JPEG

- DHT bit stealing
  - Reuse unused code words in Huffman tables
  - Lossless
  - Capacity may be zero (if Huffman tables are optimized)
- DQT bit stealing
  - Use the m LSB of the last n QT entries
  - Not lossless
  - Capacity depends on number of DQT segments
  - Capacity vs. fidelity tradeoff

DHT bit stealing results: LIVE data set and jpeg-8d



DQT bit stealing results: LIVE data set and jpeg-8d



Conclusion

- New method for compact RoI representation
  - Clearly outperforms JBIG-compressed bitmaps
  - <100 bits on average for real-world surveillance RoI data
  - Tiny overhead (2 bits) when no RoIs are present
- New lossless method to signal RoIs in JPEG images
  - Capacity >1000 bits in test data sets
  - Capacity depends on JPEG quality
  - Drawback: Does not work with optimized Huffman tables
- New lossy method to signal RoIs in JPEG images
  - Capacity >300 bits in test data sets at 75% quality
  - Capacity depends on picture size and content
  - Quality vs. capacity tradeoff allows easy adjustment

Thank you for your attention!

Questions?