**CMSE492 Lab 2. Wu and Tsai’s method**

17.03.2019

**Task:**

1. Implement Wu and Tsai’s algorithm [1] in any programming language/operating system available in CMPE-134
2. Test your implementation using Seminar 14.03.2019 examples: “Embed the following secret data, S=35, 21, 19, 42, 13, 27 (the decimals shall be converted to the binary representation without leading zeroes), into the cover image, CI=(150, 1, 1, 150, 10, 40, 40, 10, 18, 75, 75, 18) by Wu and Tsai method using ranges R1=[0,7], R2=[8,15], R3=[16,31], R4=[32,63], R5=[64,127], R6=[128,255]. Before embedding, check that the pixel pair is to be used/not used for embedding.

Extract the data embedded. Before extraction, check the pixel pair for the abandonment condition; check that the condition returns exactly the same result as it was when checking for embedding this pair.”.

1. Test your implementation on 4 host 512x512 images (Mandrill, Peppers, Jet, and Lena) and 4 secret 256x256 images (Mandrill, Peppers, Jet, and Lena) used in [1, 2]
2. For each of 16 variants of embedding secret into the covers, calculate embedding capacity, RMSE, and PSNR, and compare your results versus [1, Table 1, p. 1620; Table 2, p. 1622]. In the case of discrepancies, fix your problems, or prove that your results are correct.
3. **Defend the Lab on April 3, 2019, Wednesday, 16.30-18.20, CMPE-134 (hand in your report to Evaluator, run your program, and explain your work done).**
4. Report shall have
	1. Cover page (University, Department, Course, Semester, Year, City, Country, Lab subject, Team members, Lecturer, Lab assistant)
	2. Outline
	3. Problem definition (see items 1-4 above)
	4. Wu and Tsai’s method description
	5. Description of Wu and Tsai’s method implementation in your programming language/operating system
		1. Description of the host/secret images you use and their sources
		2. Description of the secret binary data stream obtainng
		3. Description of checking the falling-off-boundary condition using inverse difference function
		4. Description of embedding using inverse difference function
		5. Description of extraction implementation including checking the falling-off-boundary condition using inverse difference function
		6. Description of RMSE, PSNR, and embedding capacity calculation
	6. Description of the tests conducted and their results, **screenshots** of them
	7. Comparison of your results versus [1, Table 1, p. 1620; Table 2, p. 1622].
	8. Conclusion
	9. References
	10. Appendices with the code developed
	11. CD with all Lab related materials (report, images used, test results, sources, executables). CD shall be runnable (it is possible to install your program from the CD, run it on your examples, and view results you got).

**References**

1. D.-C. Wu, W.-H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letters 24 (2003) 1613–1626, <https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE492/WuPRL2003%20PVD.pdf>
2. S.-J. Wang, Steganography of capacity required using modulo operator for embedding secret image, Applied Mathematics and Computation, 164 (2005), 99-116, doi:10.1016/j.amc.2004.04.059, <https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE492/LSB%20modulo%20AMC2005.pdf>

**Grading policy: report – 50%, explanations – 50%**