2011

VŠB- TU Ostrava

Tomáš Vantuch VAN431

[PROJEKT STEGANOGRAFIA]

Steganografia pomocou LSB a úprava obrázkov pomocou DCT

Obsah

Zadanie: 2
Steganografia 3
Kompresia JPEG 4
DCT 5
Transformácia farieb z RGB (prípadne CMYK) do YC_bC_r. 5
Redukcia (podvzorkovanie) farebných zložiek. 5
Diskrétna kosínová transformácia 5
Kvantizácia DCT koeficientov 6
Huffmannovo kódovanie 7
Záver: 9

Zadanie:

Zostrojím program riešiaci steganografiu pomocou metódy LSB. Ako zapisovanú správu použijem text a vhodne komprimovaný obrázok. Ako kompresiu obrázku (v JPEG formáte) použijem DCT s Huffmannovým kódovaním.*

**použitý algoritmus na DCT a Huffmanna nie je môj vlastný ale použil som riešenie nájdené na internete. Preto sa vysvetľovaniu algoritmu nebudem venovať. Popíšem túto kompresiu ale teoreticky.*

Hrubší popis algoritmu:

Program je postavený na platforme Java s frameworkom Swing pre vykreslenie GUI.

V prvej fáze sa vyberá nosný obrázok (ktorý nesie alebo len bude niesť našu správu). Je možné si vybrať z viacerých možných formátov (JPEG, BMP, PNG) a po úspešnom otvorení je užívateľ informovaný, že obrázok bol úspešne otvorený a akú veľkú kapacitu poskytuje. V ďalšom kroku nastane :

A: Zápis do obrázku

Užívateľ do určeného políčka napíše správu a vyberie obrázok na zápis. Daný obrázok sa po otvorení skomprimuje pomocou DCT (kvalitou cca 16%- približne konštantné číslo) na menšiu veľkosť a uloží sa do súborovej zložky ako cover.jpg. Ďalej užívateľ zvolí tlačidlo pre zápis, ktorým dané dáta zapíše do zdrojového obrázku. Ako prvý krok sa zapisovaný obrázok prekonvertuje na pole bytov a spojí sa s ďalším poľom bytov, ktoré je z textovej správy. V ďalšom kroku prebehne overenie 120 skoro náhodných možností uloženia tejto správy. Vždy sa zvolí jedna z farebných zložiek, rozostup pixelov (každý prvý, druhý, tretí,...), a index prvého pixelu k zmene. Tieto možnosti sa spustia v nezávislých vláknach, kde každé zistí koľko by daná varianta musela zmeniť bitov. Všetky úspešné varianty sú uložené do kolekcie (niektoré spadnú pre nedostatok miesta napríklad) a z nich je vybraná tá, ktorá má najmenej potrebných zmien. (Rozdiel najlepšej a najhoršej je približne 15-25%). A podľa nej je zápis do obrázku vykonaný. StegoKey je vygenerovaný ako samostatný reťazec a je po úspešnom zápise predložený užívateľovi ako kľúč na dekódovanie dát.

Extrakčný kľúč sa skladá:

- *Číslo farebnej zložky*
- *Index počiatočného zmeneného pixelu*
- *Číslo udávajúce rozostup zmenených pixelov*
- *Dĺžka obrázku (zapísaného) v bytoch.*
- *Dĺžka textu (zapísaného) v bytoch.*

Výsledný obrázok aj s dátami je uložený ako stego.png.

B: Extrahovanie dát z obrázku

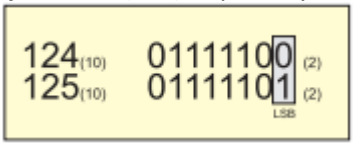
Užívateľ načíta obrázok stego.png, vloží extrakčný kľúč a program preiteruje a zloží všetky príslušné bity do obrázku a textu, ktoré nakoniec úspešne vypíše.

Steganografia

Slovo steganografia má základ v gréčtine a znamená tajné písanie. Je to veda zaoberajúca sa ukrývaním správ do iných správ. V steganografii sa používa niekoľko základných výrazov. Cover medium znamená obálku, teda médium, do ktorého sa ukrýva správa. Tá sa nazýva embedded message. Ak používame heslo, nazýva sa to stegokey. Výsledkom je potom stego-medium.

Obrázky sú ideálnym médiom na ukrývanie stegospráv. Majú obrovskú kapacitu, obrázok vo formáte BMP s rozlíšením 1024 × 768 bodov v 24-bitových farbách má veľkosť 2,25 MB.
Existujú dva typy kompresie obrázkov, a to bezstratová a stratová. Bezstratová má rovnaké množstvo údajov o obrázku pred kompresiou aj po nej. Je vhodnejšia na steganografiu a používajú ju obrázky typu BMP a GIF. Stratová kompresia dokáže ušetriť veľa miesta potrebného na uloženie obrázka, ale za cenu vypustenia časti informácií. Po kompresii teda obrázok neobsahuje toľko informácií ako pred kompresiou. Túto kompresiu používa formát JPG. Práve pre stratovú kompresiu sa tento formát menej odporúča na steganografické účely.

Obrázok si možno predstaviť ako maticu čísel, pričom každé z nich predstavuje číslo príslušnej farby. Čiernobiele obrázky majú iba jednu maticu, ktorej čísla predstavujú odtiene sivej. Farebné obrázky sú zložené akoby z troch matíc, každá pre jednu farebnú zložku, ktorými sú červená (R) zelená (G) modrá (B). Každý bod je zložený z týchto troch zložiek. Hodnota odtieňa sivej farby je v rozmedzí od 0 do 255 a takisto každá zložka R, G, B má hodnotu od 0 do 255. Dajú sa teda zakódovať do 8 bitov. Dve farby, ktoré sú v palete v tesnej blízkosti, to znamená, že ich číselná hodnota sa líši o 1, napríklad farba číslo 124 a 125, sú také príbuzné odtiene, že človek si nemusí uvedomiť ich vzájomnú zámenu v obrázku. Súčasne sa tieto farby líšia v bitovom zápise iba posledným, najmenej dôležitým bitom, ktorý sa nazýva Least Significant Bit (LSB). V praxi vyzerá binárny zápis týchto dvoch čísel takto:



Znamená to, že pre výsledný obraz je prakticky jedno, ako vyzerá LSB. Vytvára sa tak priestor na uloženie dodatočnej informácie ľubovoľného charakteru. Informácia je zapísaná v binárnom tvare, a to tak, že do každého LSB bitu každého bajtu obrázka sa zapíše 1 bit stegosprávy. Teoreticky je teda možné zapísať do obrázka stegosprávu s kapacitou 1/8 pôvodného obrázka. ***Stegokey****, teda informáciu akým postupom a koľko bitov sme do obrázku uložili, najčastejšie nejako ukryjeme priamo do obrázku pomocou nejakej funkcie, alebo jednoducho na začiatok či koniec súboru v zašifrovanom tvare. S jeho pomocou následne sme schopný skrytú správu z obrázku vyextrahovať.*

Kompresia JPEG

V rámci efektívneho využívania prenosových médií, ale aj vôbec výpočetných zdrojov bolo nutné pre viaceré typy súborov zaviesť efektívny spôsob jejich spracovania. Konkrétne obrázky typu JPEG, videá MPEG alebo hudobné súbory MP3 využívajú rôzne typy komprimačných metód pre zníženie veľkosti, avšak pri zachovaní čo najvyššej kvality výsledného komprimovaného súboru.
Zmena kvality výsledného súboru je označovaná ako chyba. Nastáva vždy u akejkoľvek stratovej kompresii a je veľmi zložito merateľná akýmkoľvek algoritmom, pretože vždy ide o výsledok, ktorý až užívateľ posúdi či daná chyba je alebo nie je veľká.

Pre príklad, v obrázku môže dôjsť k zmene jednej farebnej zložky (napr. o 10%) u 2000 pixelov, čo program odhalí a posúdi ako veľkú chybu ale ľudské oko to pritom nemusí zachytiť takže obrázok pripadá v poriadku. Naopak 5 pixelov nastavíme na bielu čo bude pre program zmena minimálna ale človeku to vo výsledku príde ako jasná vada.

Je nutné užívať teda kompresiu, ktorá súbor transformuje do čo najvernejšej podoby pre ľudské zmysly. Takýmto spôsobom bolo v roku 1989 zavedené pre formát JPEG ako základné kompresné schéma transformácia farieb, diskrétna kosínova transformácia a Huffmannovo kódovanie. Samotná komprimačná metóda je popísaná vo viacerých štandardoch a viaže sa na ňu viac ako sto patentov.

JPEG definuje štyri režimi činností, ktoré kóder i dekodér prevádza pri komprimovaní resp. dekomprimovaní. Jedná sa o sekvenčné kódovanie, ktoré je zo všetkých najčastejšie používané, pretože je najmenej náročné na pamäť a tak môže byť aplikované aj v zariadeniach ako sú digitálne fotoaparáty s malou operačnou pamäťou. Ďalej je to progresívne kódovanie, ktoré je viac náročné na výpočetné zdroje a je určené hlavne pre prenos obrázkov po sieti. A nakoniec je to bezstrátové (predikčné) – menej známe a hieratické kódovanie, ktoré podporuje mnoho rozlíšení, rýchle nahľady...
Bitová rýchlosť u JPEG je údaj o počte bitov, ktorými je popísaný pixel v obrázku. (ich počet je stanovený ako priemer, teda počet bitov na počet pixelov). U sekvenčného kódovania definujeme 4 najčastejšie bytové rýchlosti.

0.25 – 0.50 bpp	*Stredná kvalita obrazu*
0.50 – 0.75 bpp	*Dobrá kvalita, dostatočná pre väčšinu aplikácii*
0,75 – 1,50 bpp	*Vynikajúca kvalita*
1,50 – 2,00 bpp	*Nerozoznateľné od originálu*

DCT

Diskrétna kosínová transformácia sa skladá z viacerých krokov:

Transformácia farieb z RGB (prípadne CMYK) do YC_bC_r.

Jedná sa o bezstrátový typ transformácie, ktorého podstatou je oddeliť farebné zložky od zložky vyjadrujúcu jas pixelu. Samotný prevod z RGB do Y (zložka jasu) a C_b,C_r čo sú rozdielové hodnoty farieb pixelov (oproti Y) je nasledovný:

$$Y = 0{,}299\ R + 0{,}587\ G + 0{,}114\ B$$

$$Cb = -0{,}1687\ R - 0{,}3313\ G + 0{,}5\ B + 128$$

$$Cr = 0{,}5\ R - 0{,}4187\ G - 0{,}0813\ B + 128$$

Pre spätný prechod:

$$R = Y + 1.402\ (Cr-128)$$

$$G = Y - 0.34414\ (Cb-128) - 0.71414\ (Cr-128)$$

$$B = Y + 1.772\ (Cb-128)$$

Redukcia (podvzorkovanie) farebných zložiek.

Ľudské oko je vybavené väčším počtom receptorov, ktoré vnímajú svetlosť ako samotnú farbu. Preto zmena farebných zložiek nemusí byť tak razantne vnímaná ako zmena jasu. Redukcia farebných zložiek spočíva v spracovaní C_b, a C_r zložiek a to jednoduchým spriemerovaním buďto susedných dvoch alebo štyroch pixelov. Pri podvzorkovaní dvoch pixelov dostaneme namiesto 6 bytov len 4, čo je úspora až 66% a pri podvzorkovaní štyroch pixelov (tvoriacich štvorec) získame namiesto 12 bytov len 6, čo je úspora až 50%. Je treba si ale uvedomiť, že sa jedná o stratový prevod, pretože ďalej nemáme informácie o farbe jednotlivých pixelov.

Diskrétna kosínová transformácia

Tento samotný krok je robený nad zložkou Y každého pixelu a to prepočtom pomocou kvantizačnej funkcie čím získame koreláciu medzi susednými (prípadne vzdialenejšími) pixelmi – tzv. medzipixelová redundancia.
DCT je rozlišované v jednorozmernej úrovni pre určité signálny v systéme jednorozmerného poľa ako 1D DCT :

$$t(k) = c(k) \sum_{n=0}^{N-1} s(n) \cos \frac{\pi(2n+1)k}{2N}$$

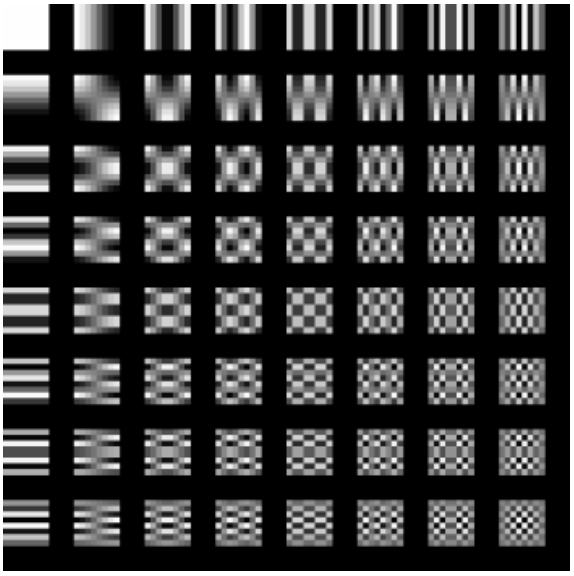
Z tohto vzťahu bázová funkcia pre 1D DCT: $\cos \frac{\pi(2n+1)k}{2N}$

Pre dvojrozmerný raster (obrázky) hovoríme o dvojrozmernej diskrétnej kosínovej transformácii:

$$t(i,j) = c(i,j) \sum_{n=1}^{N-1} \sum_{m=0}^{N-1} s(m,n) \cos \frac{\pi(2m+1)i}{2N} \cos \frac{\pi(2n+1)j}{2N}$$

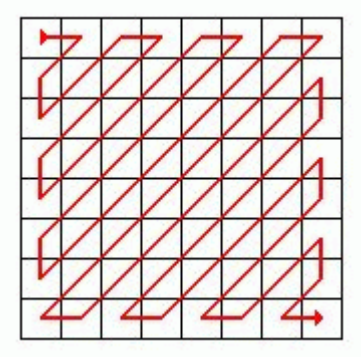
Bázová funkcia pre 2D DCT má tvar: $\cos \frac{\pi(2m+1)i}{2N} \cos \frac{\pi(2n+1)j}{2N}$

A typický priebeh 2D DCT pre N=8



Kvantizácia DCT koeficientov

Kvantizačná tabuľka je matica koeficientov nastavená (zväčša) podľa štruktúry obrázku a použitej kvality kompresie. Pre vyzdvihnutie vysokých frekvencií (hrany, šum) sa v kvantizačnej tabuľke vyskztujú koeficienty s vysokými hodnotami (v konkrétnom, úzkom zastúpení), a pre nižšie frekvencie (priestor s vyrovnanými farebnými zložkami) sú kvantizačné koeficienty nižšie. Pomocou vhodne zvolenej kvantizačnej tabuľky môžeme presnejšie vygenerovať vodorovné alebo zvislé hrany, či iné ostré prekrytia farieb. Pri vyššej stratovosti (nevhodne zvolená kv. tabuľka, pre zmenšenie veľkosti výsledného súboru) získame kvantizáciou namiesto hrán šum. Takže výsledné hodnoty po DCT podelíme hodnotami v kvantizačnej tabuľke (vždy jeden s daným, ktorý mu prislúcha), v tomto prípade sa jedná o celočíselné delenie (nastáva ďalší stratový jav).



Po prevedení kvantizácie ďalej spracujeme DC zložky, keď od DC zložky v každom bloku odčítame túto zložku z predošlého bloku. Keďže sa jedná o odčítanie stejnosmernej zložky, výsledkom bude rad nízkych čísel(núl).

V ďalšom kroku, v ktorom preskladáme maticu 8x8 pomocou cik-cak sekvencie na lineárny tvar, dostaneme väčšinu núl za seba takže ich môžeme veľmi efektívnym spôsobom zakódovať.

Huffmannovo kódovanie

Je princíp zápisu dát pomocou kódových slov, systémom v ktorom majú najčastejšie sa vyskytujúce slová najkratšiu bitovú dĺžku. Pre jednotlive zložky obrazu JPEG (DC a AC) sú použité kódovacie tabuľky, pomocou ktorých sa s využitím vlastností Huffmannova kódovania šetrí ďalšia kapacita.

Vo vkladanom čísle rozlíšime o akú zložku ide (DC/AC), následne podľa jeho veľkosti určíme diferenčnú kategóriu a nakoniec podľa toho či sa jedná o luminanciu (Y) alebo chrominanciu (Cb - Cr) vyberieme kódové slovo.

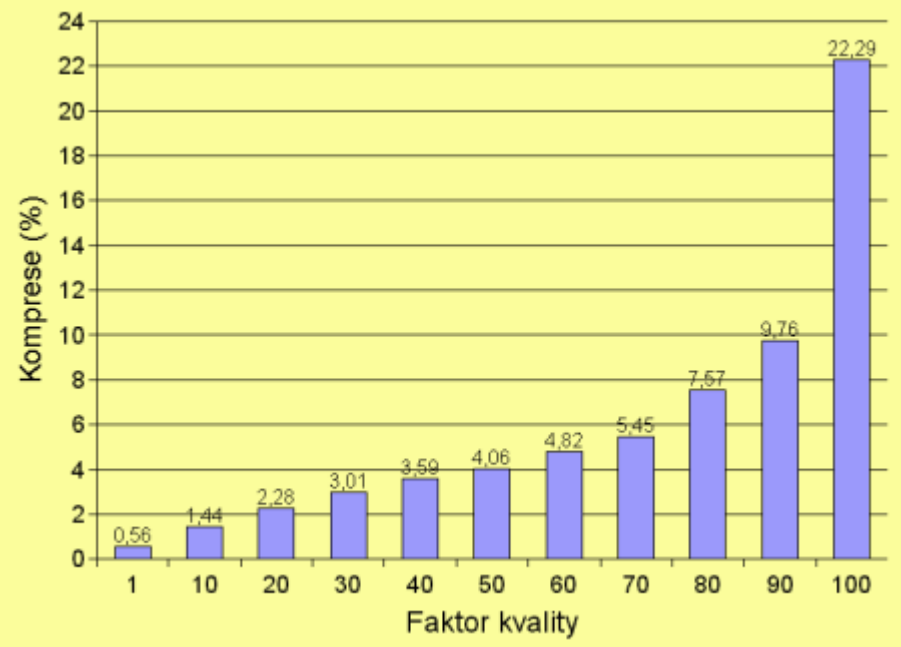
AC zložky pre vysoké frekvencie majú po kvantizácii väčšinou nulové hodnoty a cik-cak zápisom sa nachádzajú na konci bloku 8x8, preto ich vyjadrovať nemusíme (nieje pre ne ani diferenčná kategória) ale blok ukončíme značkou EOB.

Tabuľky diferenčných kategórií pre DC zložky.

Diferenční kategorie	Rozsahy záporných hodnot	Rozsahy kladných hodnot
0	–	0
1	–1	1
2	–3,–2	2,3
3	–7..-4	4..7
4	–15..-8	8..15
5	–31..-16	16..31
6	–63..-32	32..63
7	–127..-64	64..127
8	–255..-128	128..255
9	–511..-256	256..511
10	–1023..-512	512..1023
11	–2047..-1024	1024..2047

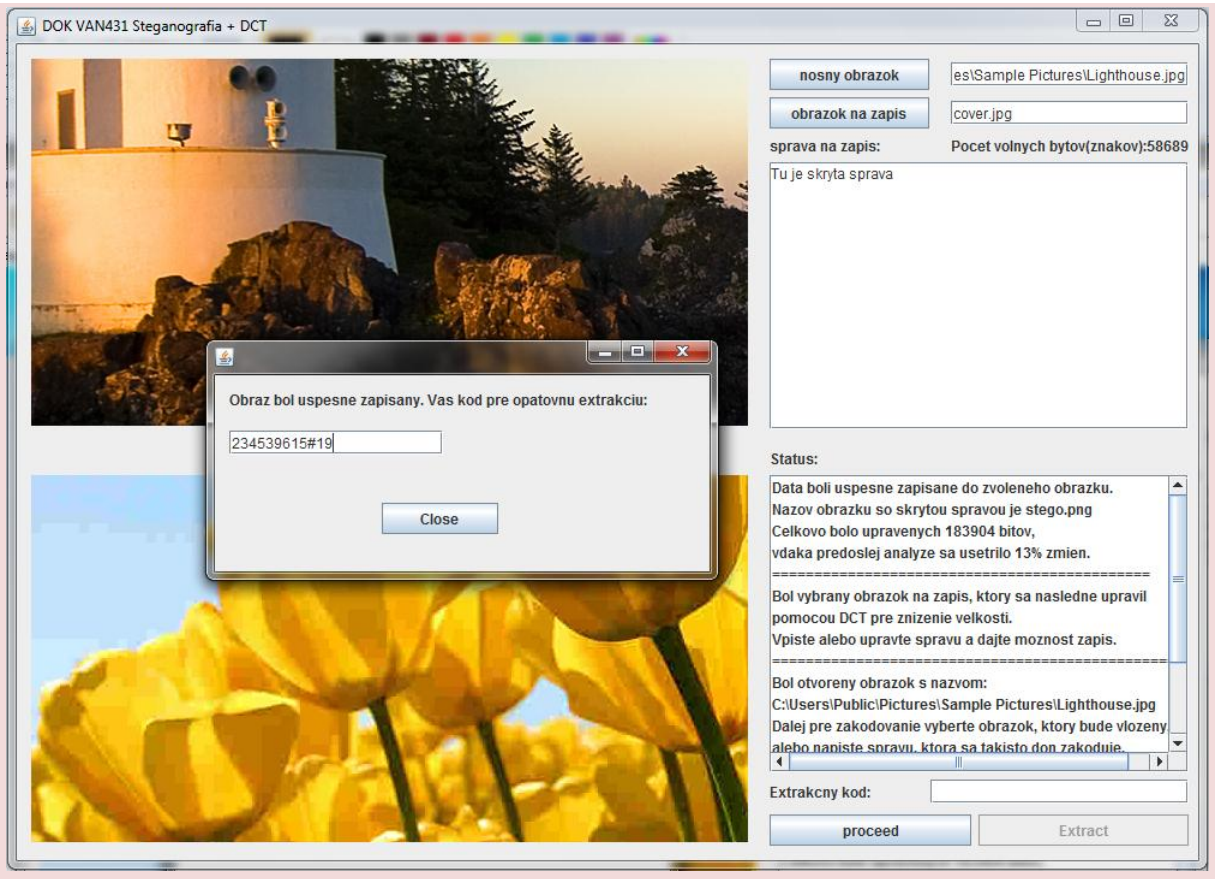
Diferenční kategorie	Lum	Luminance (codeword)	Chr	Chrominance (codeword)
0	2	00	2	00
1	3	010	2	01
2	3	011	2	10
3	3	100	3	110
4	3	101	4	1110
5	3	110	5	11110
6	4	1110	6	111110
7	5	11110	7	1111110
8	6	111110	8	11111110
9	7	1111110	9	111111110
10	8	11111110	10	1111111110
11	9	111111110	11	11111111110

Pomer kvalita k veľkosti zapisovaného obrázku (DCT kompresia):



(zdroj root.cz článok JPEG - král rastrových grafických formátů)

Ukážka programu:



Záver:

Formou tohto projektu som si vyskúšal problematiku steganografie. LSB metóda patrí dnes už medzi zastarané a ľahko prekonateľné, stále je možný priestor k jej využitiu. Takisto som si teoreticky osvojil diskrétnu kosínovú transformáciu s Huffmannovým kódovaním ako jednu z najpožívanejších kompresných metód.

Projekt neposkytuje žiadne zásadné výsledky v obore, len ukazuje jednu z možností ako môže byť takáto problematika riešená.

Zdroje:

- *http://sk.wikipedia.org/*
- *http://www.root.cz/*
- *http://www.guillermito2.net/stegano/*
- *A ďalšie…*