



**ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ**

**ПРЕДИКТИВНО КОДОВАЊЕ
У КОМПРЕСИЈИ 3D СТ СНИМАКА**

Студенти: Наташа Кузмановић, Славица Савић, Александар Пајкановић

Предметни наставник: др Зденка Бабић
Предметни асистент: Владимир Рисојевић

Број бодова:

Бања Лука
Јуни, 2009. године

ПРЕДИКТИВНО КОДОВАЊЕ У КОМПРЕСИЈИ 3D СТ СНИМАКА

Наташа Кузмановић, Славица Савић, Александар Пајкановић

E-mail: barjaktar@gmail.com

1. ДЕФИНИСАЊЕ ПРОЈЕКТА

Иако је капацитет меморија резервисаних за складиштење података од интереса стално у порасту, као и брзина којом се ти подаци преносе, и даље постоји проблем великог броја података које је потребно обрадити у кратком временском року. Када се ради о дигиталним сликама, у неким областима примјене могуће је извршити нека занемаривања и тиме допринијети брзини и једноставности обраде. Међутим, у случају сивих медицинских слика насталих компјутерском томографијом и магнетном резонансом, занемаривање и најситнијих детаља није дозвољено. Стога се проблему приступа на друге начине, од којих ћемо један проучити у овом раду.

2. СПЕЦИФИКАЦИЈА

2.1. Спецификација софтвера

Операције (образложене у даљем тексту) над сликама аутори су вршили у програмском пакету MATLAB®, користећи његове стандардне функције, али и оне из модула Digital Image Processing.

2.2. Спецификација хардвера

Програм, настао вршењем поменутих операција, аутори су извршавали на платформи следећих перформанси:

Intel® Celeron® M processor 1.50 GHz

512MB of RAM

AMD® Turion® 3.2GHz

3 GB of RAM

3. ОПИС ДИЗАЈНА

3.1. Увод

Програм ради у два корака. Први је предикција, а други кодовање.

Операција предикције има смисла да се врши над дигиталним сликама само уколико постоји јака корелација између пиксела те слике. Медицинске слике имају ту особину, тако да се ова операција примјењује успјешно. Сврха предикције, као што и сама ријеч каже, је да предвиди вриједности. У овом случају то су вриједности свјетлине пиксела слике, а

предвиђање се обавља тако што се претпоставља нека свјетлина (на основу свјетлина сусједних пиксела – користимо особину јаке корелације), да би се затим мјерила грешка, односно разлика стварне и предвиђене свјетлине. Тако добијамо мале апсолутне вриједности свјетлине за чије је складиштење потребно мање меморијског простора.

Како бисмо још више смањили потребе за меморијским простором, вршимо и кодовање тако умањених вриједности.

3.2. Дијаграми тока извршавања

Већ је речено да се програм извршава у два корака, која су приказана графички на дијаграму тока.



Прикажимо графички и обрнут поступак, тј. начин на који на пријему од добијених података реконструирамо слике над којима су извршене поменуте операције.



4. ИМПЛЕМЕНТАЦИЈА

4.1. Увод

Раније је споменуто да се програм састоји из двије цјелине. Погледајмо сада детаљније о чему се заправо ради и проучимо алгоритам чија је реализација настала програм. Детаљи самог кода, односно саме примјене алгоритма, се налазе у приложеној датотеци.

4.2. Реализација

JPEG је тренутно најчешће кориштени начин компресије слика. Међутим, оно што је лоша страна код овог метода је чињеница да у својој реализацији користи дискретну косинусну трансформацију, дакле захтјева прилично комплексне прорачуне са великим бројем основних операција. Иако постоје брзи алгоритми за рачунање ових трансформација, као и за дискретну Фуријеову трансформацију, ово ипак знатно увећава сложеност самог поступка, али и повећава захтјеве за хардверским ресурсима.

За операцију предикције, аутори су користили следеће формуле:

$$d_x(n_1, n_2, n_3) = \begin{cases} x(n_1, n_2, n_3) - x(n_1 - 1, n_2, n_3), & 0 < n_1 < l_x; \\ x(n_1, n_2, n_3), & n_1 = 0; \end{cases}$$

$$d_y(n_1, n_2, n_3) = \begin{cases} d_x(n_1, n_2, n_3) - d_x(n_1, n_2 - 1, n_3), & 0 < n_2 < l_y; \\ d_x(n_1, n_2, n_3), & n_2 = 0; \end{cases}$$

$$d_z(n_1, n_2, n_3) = \begin{cases} d_y(n_1, n_2, n_3) - d_y(n_1, n_2, n_3 - 1), & 0 < n_3 < l_z; \\ d_y(n_1, n_2, n_3), & n_3 = 0; \end{cases}$$

гдје је $x(n_1, n_2, n_3)$ улазни сигнал, тј. слика; са $d_x(n_1, n_2, n_3)$, $d_y(n_1, n_2, n_3)$ и $d_z(n_1, n_2, n_3)$ су означене израчунати сигнали грешке по осама x , y и z , респективно.

Обратимо пажњу на чињеницу да су слике (које се, иначе, појединачно, представљају дводимензионим матрицама), да би се овај алгоритам примјенио, сложене „једна иза друге“. Другим ријечима, креирана је тродимензиона матрица у коју су учитане слике тако да трећа димензија представља редни број слике, односно слоја.

Из једначина се такође види да се предикција обавља на основу три димензије. Дакле, на претпостављену вриједност неког пиксела утичу три пиксела, и то:

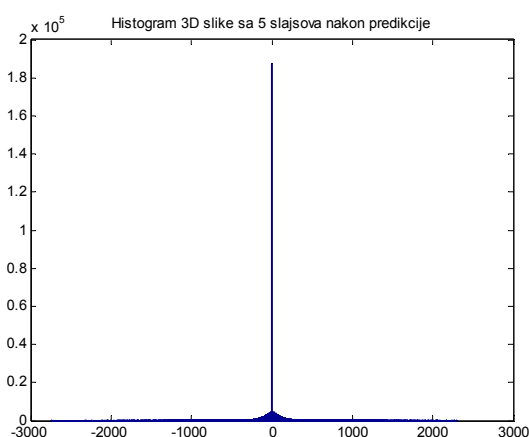
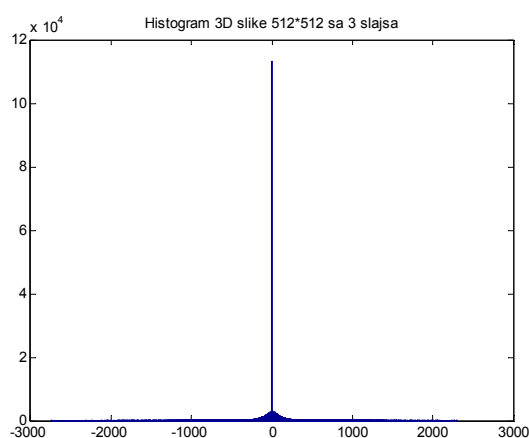
- претходни по хоризонталној оси;

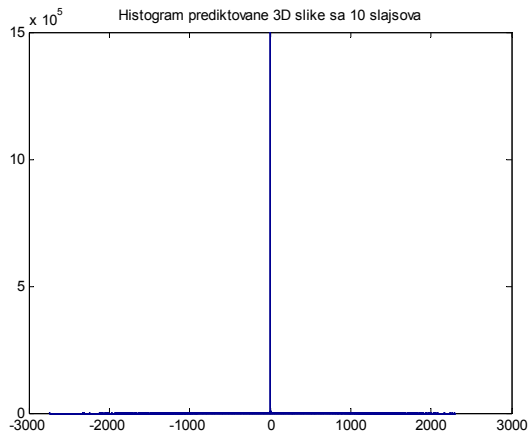
- претходни по вертикалној оси;

- претходни по „дубини“, односно слоју, што је у ствари пиксел који на претходној слици има исте координате као посматрани.

Изузеци су пиксел на координатама (0,0,0), који се не предиктује (немамо на основу чега да га претпоставимо) и пиксели који су први у првој колони слике (предвиђају се само на основу вриједности одговарајућег пиксела са претходне слике), као и пиксели који су први у осталим колонама (предвиђају се на основу два пиксела – одговарајућег из претходне колоне на истој слици и одговарајућег из претходне слике).

Након примјењене предикције, добијамо слике чији пиксели имају свјетлине јако блиске нули. То ћемо најбоље уочити ако посматрамо хистограм излазне слике, који ћемо посматрати у три случаја. У првом смо одједном обрађивали три слике, у другом пет, а у трећем десет. Иако ће истовремена обрада већег броја слика конзумирати много више процесорског времена и радне меморије, добра страна је што ће већи проценат пиксела имати вриједност свјетлине у околини нуле. Узрок ове појаве је чињеница да повећање броја слика у тродимензионој матрици појачава и корелацију, па ће операција предикције дати још боље резултате.





Од изузетне важности за разумијевање поступка је напоменути да излазни сигнал при операцији предикције није предикована слика, него сигнал грешке који представља разлику између тачних вриједности свјетлина пиксела и оних које ћемо претпостављати при операцији депредикције.

Алгоритам којим је обављена предикција је имплементиран у приложеној датотеци под називом `predikcija.m`.

Послије операције предикције, као што смо већ видјели, слиједи операција кодовања. Сада треба одабрати метод кодовања. Посматрајући хистограм и уочавајући да је огромна већина пиксела концентрисана на вриједности нула и у њеној околини, закључујемо да ће Хафманово кодовање пружити добре резултате. Ова операција је вршена примјеном готових функција преузетих из књиге *Digital Image Processing Using MATLAB®*, с тим да је на једној од њих извршена модификација која је омогућила функционисање цијелог процеса.

За потребе овог рада функција која врши поменути операцију је морала бити модификована. С обзиром да су аутори при извршавању програма и симулација располагали са ограниченом опремом, код је морао бити оптимизован на следећи начин: умјесто да се у исто вријеме врши кодовање свих симбола који се налазе у рјечнику, кодовање се врши дио по дио. Вектор у којем је садржан рјечник је издијељен на двјестопедесетшест подвектора, те је затим на сваки од њих понаособ примјењена операција кодовања, послједње је кодовани подвектор додат претходном. Тако на крају, послједње кодовања двјестопедесетшест појединачних подвектора, имамо исти резултат као да су сви симболи кодовани одједном.

Модификована верзија функције `mat2huff` (која је преузета из књиге *Digital Image Processing Using MATLAB®*) се налази у приложеној датотеци под називом `hkod.m`.

На дијаграму тока смо већ видјели да се, за потребе реконструкције слика, на пријему обављају инверзне радње инверзним редослиједом, дакле прво декодовање, а затим и депредикција.

Декодовање примјеном функције `huff2mat` је успјешно извршено на доступним платформама, стога није било потребе за модификацијама кода. Ова функција је такође преузета из књиге *Digital Image Processing Using MATLAB®* и приложена уз рад.

Математички операцију депредикције представљамо следећим једначинама:

$$d_y(n_1, n_2, n_3) = \begin{cases} d_y(n_1, n_2, n_3 - 3) + d_z(n_1, n_2, n_3), & 0 < n_3 < l_z; \\ d_z(n_1, n_2, n_3), & n_3 = 0; \end{cases}$$

$$d_x(n_1, n_2, n_3) = \begin{cases} d_x(n_1, n_2 - 1, n_3) + d_y(n_1, n_2, n_3), & 0 < n_1 < l_y; \\ x(n_1, n_2, n_3), & n_2 = 0; \end{cases}$$

$$x(n_1, n_2, n_3) = \begin{cases} x(n_1 - 1, n_2, n_3) + d_x(n_1, n_2, n_3), & 0 < n_1 < l_x; \\ d_x(n_1, n_2, n_3), & n_1 = 0; \end{cases}$$

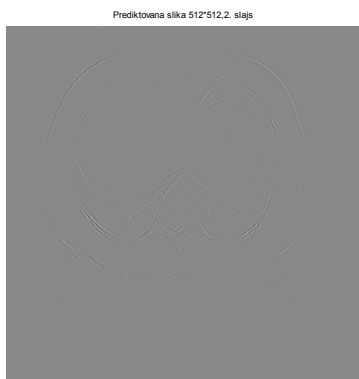
Тек овдје ми вршимо праву предикцију, односно предвиђање вриједности свјетлина пиксела, које затим познатим сигналом грешке (који смо добили декодовањем) коригујемо како бисмо добили тачне вриједности свјетлина појединих пиксела и у потпуности реконструисали почетне слике. Имплементација алгоритма депредикције садржана је у приложеној датотеци `depredikcija.m`.

Све ове функције су обједињене у једној датотеци чијим извршавањем пролазимо кроз цијели поступак, под називом `sve.m`. Детаљи о покретњу исте дати су у самој датотеци.

4.3. Резултати симулације

Упоредимо оригиналну слику (потпуно је иста као и реконструисана слика, па нећемо два пута приказивати исто) и оно што се добије послједње примјењене предикције по наведеној формули.

Напоменимо да су пиксели слике добијене предикцијом у ствари већином црни, а они који нису, јако су блиски црној боји. Када бисмо је приказали онако како заиста изгледа, не бисмо уочили сигнал грешке. Због тога је приказујемо са развученим хистограмом – одатле нијансе сиве боје.



Имајући у виду раније приказани хистограм добијамо очекиван резултат.

Колико год нам биле битне апсолутне вриједности степена компресије, много ћемо бољу слику о значају предикције имати уколико тако добијени степен компресије упоредимо са степеном компресије добијеним на други начин. Посматрајмо табелу у којој су приказани степен компресије постигнути кориштењем поступка описаног у досадашњем тексту и само Хафмановим кодовањем. Параметар који ћемо варирати је број слика које обрађујемо одједном, што је објашњено у поглављу 4.2. Такође, посматрајмо и однос добијених степена компресије на ова два начина. Резултати су добијени као излаз приложене функције `imratio.p`.

Број одједном обрађених слика	3	5	10
предикција и Хафманово кодовање	4.3530	4.3619	4.4590
Хафманово кодовање	1.9828	1.9659	1.9448
однос степена компресије	2.1954	2.2188	2.2928

Предност поступка који укључује предикцију је очигледна – видимо да је компресија на тај начин постигнута до, приближно, 2.3 пута боља. Са порастом броја истовремено обрађених слика при поступку који примјењује само Хафманово кодовање степен компресије опада, док при поступку који укључује предикцију прије кодовања, степен компресије расте. Тако је однос степена компресије постигнутог при обради десет слика и оног који добијамо обрадом три слике 1.0244, што представља пораст од скоро 2.5%.

5. КОМЕНТАРИ

Закључак који се намеће је да је алгоритам описан и примјењен у овом раду изузетно једноставан и учинковит. Без кориштења комплексних математичких операција успијева да постигне резултате који су бољи од оних које постижу знатно компликованији, комерцијални алгоритми. Дакле, постиже се уштеда како финансијских тако и хардверских ресурса.

Занимљиво за примјетити је и да слике које се обрађују на овај начин нису ограничене, што се тиче димензија. Ради се о томе да су, због математичких операција које користе, неки алгоритми ограничени на обраду слика чије су димензије степен броја два. Овдје то није случај, јер се не користе брзи алгоритми за рачунање брзе Фуријеове или дискретне косинусне трансформације.

Иако су за потребе рада кориштене слике настале компјутерском томографијом (СТ), исти алгоритам је могуће примјењивати и за слике добијене другим уређајима који се користе у медицинске сврхе.

Аутори су у току израде рада посједовали серију од десет слика са јаким корелацијом, те су били ограничени на истовремено обрађивање максимално толиког броја слика. Посматрањем резултата закључујемо да тај број игра велику улогу у постигнутом степену компресије, те да би при обради серије већег броја слика степен компресије био још већи. У ствари, једино ограничење представљају хардверски ресурси.

Алгоритам је могуће уопштити тако да врши предикцију вишег реда, што се огледа у томе да се вриједност свјетлине пиксела не претпоставља на основу по једног претходног по свакој од димензија, него на основу више њих по свакој од димензија. Могуће је да се оваквим поступком добије још већи степен корелације, а тиме и боља компресија, али ће довести до усложњавања алгоритма и веће захтјевности хардверских ресурса.

6. ЛИТЕРАТУРА

- [1] *Lossless Volumetric Medical Image Compression with Progressive Multi-planar Reformatting Using 3D DPCM*, Vishram Nandedkar, S.V. Bharath Kumar and Sudipta Mukhopadhyay
- [2] *Digital Image Processing Using MATLAB ®*, Gonzales, Woods, Eddins, 2004

7. ПРИЛОЗИ

Уз овај рад приложене су следеће датотеке:

- ucitavanje.m;
- predikcija.m;
- Hkod.m;
- huffman.p
- unravel.dll
- huff2mat.p;
- depredikcija.m;
- imratio.p;
- sve.m;
- серија од десет слика кориштених за тестирање алгоритма.