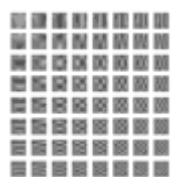


Zovšeobecnená DCT obrazových segmentov pomocou Householderovho algoritmu

Gladišová Iveta · Informačné technológie

18.04.2016



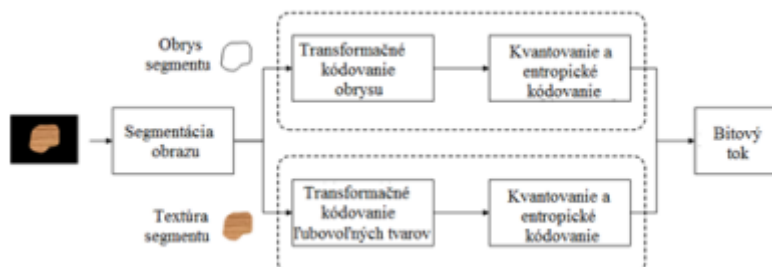
Článok sa zaoberá aplikáciou Householderovho algoritmu ortogonalizácie bázových funkcií zovšeobecnenej diskkrétnej kosínusovej transformácie obrazových segmentov. Testovanie tohto algoritmu ortogonalizácie sme uskutočnili na syntetizovaných a tiež na reálnych obrazových segmentoch. Householderov algoritmus ortogonalizácie vypočíta všetky ortogonálne bázové funkcie naraz, je veľmi presný, vykazuje výrazne malú (takmer nulovú) strednú kvadratickú chybu pre jednotlivé testované obrazové segmenty. Nevýhodou Householderovho algoritmu je najmä časová náročnosť výpočtu, ktorá so zväčšujúcimi sa blokmi v obrazovom segmente rastie. Je však vhodnejší v porovnaní s Gram-Schmidtovým algoritmom ortogonalizácie bázových funkcií, ktorý počíta s postupnou ortogonalizáciou a v niektorých prípadoch sa vyznačuje nestabilitou, najmä pri veľkých veľkostiach blokov, čím môže dôjsť k strate ortogonalít medzi bázovými funkciami a tým k vytváraniu neortogonálnej bázy.

1. Úvod

Jedným zo základných nástrojov kompresie dát pri kódovaní videa je transformácia obrazu pomocou diskkrétnej kosínusovej transformácie (DCT), ktorá sa používa v medzinárodných štandardoch ako MPEG 1, MPEG 2, MPEG-4, H.261 a H.263 [1 – 3]. Klasická DCT používaná v štandardných videokodkoch prináša so sebou aj niekoľko nevýhod. Jednou z nich je zhoršená reprodukcia obrysu vizuálneho objektu. Kódovanie obrazu však využíva veľké množstvo transformačných algoritmov kódovania. Medzi tieto patria aj blokovo-orientované alebo oblastne-orientované obrazové kódovanie. Princíp blokovo-orientovaného kodéra (JPEG, H.261, MPEG) spočíva v rozčlenení obrazu do pravouhlých blokov obrazových prvkov (op) a kódovanie každého bloku osobitne ([7], [8], [9]), čo však môže spôsobiť zviditeľnenie hraníc blokov v rekonštruovanom obraze, nazývané tiež blokový efekt (zvlášť pri nízkych bitových rýchlostiach).

Väčšina týchto algoritmov nevyžíva interblokovú koreláciu, čo je ich nevýhodou. V oblastne-orientovanom obrazovom kódovaní [5], [6], [11] sú samostatne kódované hranice segmentov v jednotlivých blokoch. Vzhľadom na potrebu prenášať tvar segmentu [19] nemusí byť oblastne-orientovaný kodér nevyhnutne účinnejší ako tradičný blokovo-orientovaný kodér. Ale v prípade výkonnej reprezentačnej schémy segmentu (t.j. jeho tvaru aj obsahu) je možné dosiahnuť ďalšiu kompresiu údajov a

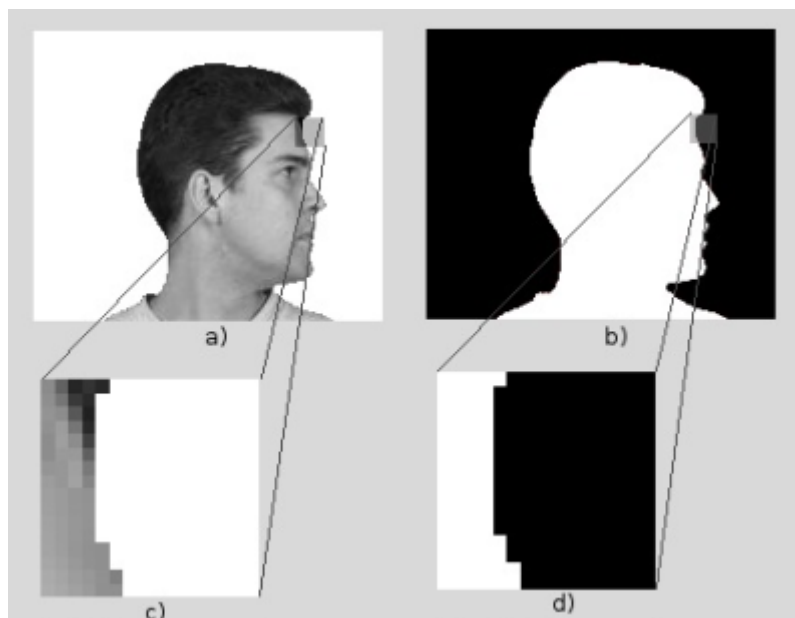
lepšiu kvalitu obrazu pri transformačnom kódovaní [12], [13]. Bloková schéma takého systému transformačného kódovania obrazu je na obr. 1. Zhoršená reprodukcia segmentov vizuálnych objektov je jednou z nevýhod pri použití klasickej DCT [10], [4]. Algoritmus DCT s ortogonalizáciou jej bázových funkcií pre dané segmenty pomocou Gram- Schmidtovej [15, 16] alebo Householderovej metódy [4] umožňuje lepšiu reprodukciu obrazových segmentov.

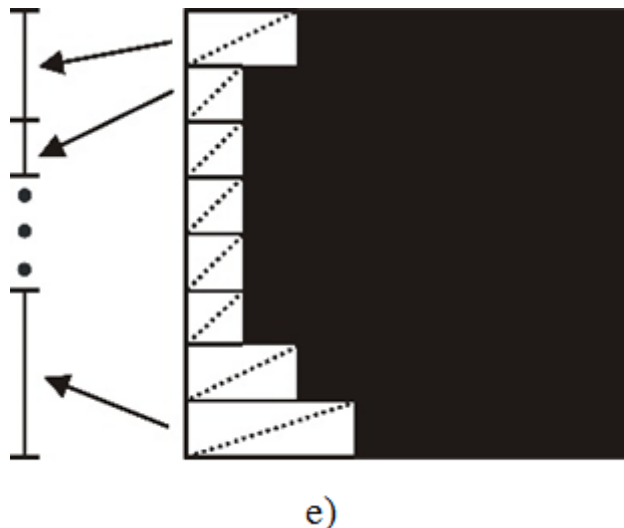


Obr. 1. Bloková schéma princípu účinného transformačného kódovania obrazových segmentov.

2. Householderov algoritmu ortogonalizácie bázových funkcií DCT

Algoritmus DCT na skvalitnenie reprodukcie obrazových segmentov pri jeho spätnej transformácii je založený na poznaní binárneho tvaru vizuálneho objektu [17]. Na obr. 2 je zobrazený binárny tvar vizuálneho objektu daného vstupného obrazu, výber jedného z jeho blokov so segmentom a usporiadanie hodnôt op segmentu do vektora podľa tvaru tohto segmentu v danom bloku.





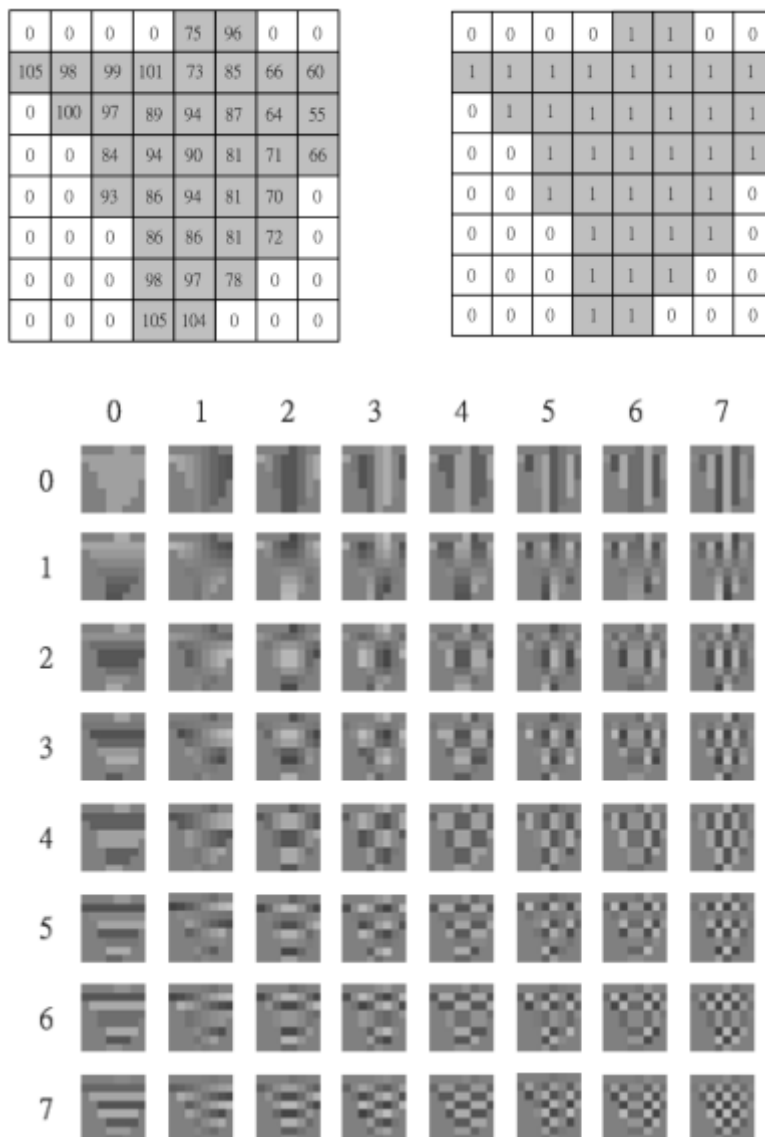
Obr. 2. a), c) Vstupný obraz a výber bloku so segmentom; b), d) výber bloku so segmentom z binárneho tvaru vizuálneho objektu; e) usporiadanie hodnôt op segmentu do vektora podľa tvaru segmentu v bloku.

Podľa toho, koľko je a ako sú rozložené op v segmente daného bloku, je možné vytvárať nové bazové funkcie DCT. Podľa tvaru segmentu sa z bazových ortogonálnych funkcií klasickej DCT vyberú iba tie hodnoty, ktoré sú na tých istých pozíciách ako op segmentu v bloku. Vyberajú sa po riadkoch od ľavého okraja segmentu po pravý okraj, zhora dole a usporiadajú sa do vektorov, ako je znázornené na obr. 2e. Takto získaný vektor je vlastne nová bazová funkcia, pričom sa vytvorí len toľko bazových funkcií, koľko je op v segmente. Tieto bazové funkcie však nie sú ortogonálne, preto sa musia ortogonalizovať. Tým, že je známy tvar segmentu op v bloku, tak vieme pre ich rôzne tvary vytvárať im zodpovedajúce systémy ortogonálnych bazových funkcií. Potom pomocou nich možno vykonať DCT obrazových segmentov v daných blokoch.

Vo všeobecnosti platí, že ak skalárny súčin dvoch funkcií $\mathbf{b} = b(x_i, y_i)$ a $\mathbf{d} = d(x_i, y_i)$ je rovný nule (t.j. $\langle \mathbf{b}, \mathbf{d} \rangle = 0$), tak funkcie nazývame ortogonálne. Aby sme získali vlastnosť ortogonalít s ohľadom na tvar segmentu, musíme použiť do funkcií súradnice iba tých dvojíc (x_i, y_j) op, ktoré sa nachádzajú vo vnútri segmentu. Iná možnosť je tá, že použijeme váhovú funkciu $v(x, y)$, ktorá je mimo segmentu rovná nule a vo vnútri jednej (obr.3 vpravo)

$$(\mathbf{b}, \mathbf{d}) = \sum_x \sum_y v(x, y) b(x, y) d(x, y) \quad (1)$$

Je zrejmé, že nové funkcie (vektory) vytvorené z pôvodnej ortogonálnej bázy DCT hodnotami op nachádzajúcimi sa na pozíciách určených tvarom segmentu nemôžu byť ortogonálne, keď zoberieme do úvahy viaceré rôzne tvary týchto segmentov v obraze. Segmenty sa odlišujú nielen svojím tvarom, ale aj počtom op, ktoré obsahujú. Preto je nevyhnuté nájsť špecifické ortogonálne bazové funkcie pre každý jeden odlišný segment. Ortogonalizáciu bazových funkcií vykonáme Householderovým algoritmom, na ktorý sme sa bližšie zamerali. Na obr.3 vľavo je ukážka matice segmentu vybraného bloku s reálnymi hodnotami jeho op, vpravo je jej váhovaná verzia (binárne hodnoty op) a dole je systém bazových dvojrozmerných kosínusových funkcií vzhľadom na daný tvar segmentu.



Obr.3 Vľavo - ukážka matice reálneho obrazového segmentu doplnená nulovými hodnotami všade tam, kde sú op mimo segment; vpravo - matica jeho váhovanej verzie (binárna reprezentácia obrazového segmentu); dole - bázové obrázky vzhľadom na daný tvar segmentu [18].

Vo všeobecnosti uvažujme maticu \mathbf{R} (horná trojuholníková matica pozostávajúca z ortogonálnych koeficientov) a maticu \mathbf{U} o rozmere $n \times n$ pozostávajúcu z n lineárne nezávislých bázových funkcií \mathbf{u}_k , každá je daná n - bodmi odobranými z pôvodnej ortogonálnej bázy DCT na miestach zodpovedajúcim tvaru segmentu ([14], [15]). Householderov algoritmus používa nasledujúcu spôsob ortogonalizácie, pri ktorom musí platiť, že $\mathbf{PU}=\mathbf{R}$. Chceme, aby matica \mathbf{P} o rozmere $n \times n$ pozostávala z ortogonálnych bázových funkcií (zohľadňujúcich tvar segmentu). Táto matica sa nazýva Householderovou maticou. Začneme najprv s Householderovou maticou \mathbf{P}_1 s rozmerom $n \times n$, pre ktorú platí [4]

$$\mathbf{P}_1 = \mathbf{E} - 2\mathbf{w}\mathbf{w}^T \quad (2)$$

s jednotkovou maticou \mathbf{E} a normovaným stĺpcovým vektorom \mathbf{w} ($\mathbf{w}^T\mathbf{w}=1$). \mathbf{P}_1 je symetrická a ortogonálna matica, ako to vidno z týchto nasledujúcich dvoch rovníc

$$\mathbf{P}_1^T = \mathbf{E}^T - (2\mathbf{w}\mathbf{w}^T)^T = \mathbf{E} - 2(\mathbf{w}^T)^T \mathbf{w}^T = \mathbf{E} - 2\mathbf{w}\mathbf{w}^T = \mathbf{P}_1 \quad (3)$$

$$\begin{aligned} \mathbf{P}_1^T \mathbf{P}_1 &= \mathbf{P}_1 \mathbf{P}_1 = (\mathbf{E} - 2\mathbf{w}\mathbf{w}^T)(\mathbf{E} - 2\mathbf{w}\mathbf{w}^T) = \\ &= \mathbf{E} - 2\mathbf{w}\mathbf{w}^T - 2\mathbf{w}\mathbf{w}^T + 4\mathbf{w}\mathbf{w}^T = \mathbf{E} \end{aligned} \quad (4)$$

Pre špeciálny vektor \mathbf{w} , ktorého odvodenie ukážeme neskôr, dostaneme maticu \mathbf{P}_1 , ktorá umiestňuje prvý stĺpec matice \mathbf{U} do stĺpcového vektora len s jedným nenulovým prvkom na prvej pozícii. Pre prvý stĺpec matice \mathbf{U} t.j. vektor \mathbf{x} a prvý stĺpec jednotkovej matice \mathbf{E} t.j vektor \mathbf{e}_1 môžeme napísať

$$k\mathbf{e}_1 = \mathbf{P}_1 \mathbf{x} \quad (5)$$

$$\begin{bmatrix} k \\ 0 \\ \dots \\ 0 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

pričom $k = \|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$ a má opačné znamienko ako i -tá zložka vektora \mathbf{x} , kde x_i je prvá súradnica vektora \mathbf{x} a vektor $\mathbf{e}_1 = (1, 0, 0, \dots, 0)^T$. Aplikovaním \mathbf{P}_1 na $\mathbf{U}^{(0)} = \mathbf{U}$ dostaneme novú maticu $\mathbf{U}^{(1)}$

$$\mathbf{U}^{(1)} = \mathbf{P}_1 \mathbf{U}^{(0)} = \begin{bmatrix} k & * & \dots & * \\ 0 & & & \\ \dots & \mathbf{P}'_2 & & \\ 0 & & & \end{bmatrix} \quad (6)$$

Prvý riadok tejto matice $\mathbf{U}^{(1)}$ je aj prvým riadkom požadovanej matice \mathbf{R} a prvý stĺpec obsahuje nuly pod prvým prvkom. Teraz zostrojíme maticu \mathbf{P}'_2 , ktorá je menšia o jeden riadok (o prvý) a jeden stĺpec (prvý) oproti \mathbf{P}_1 , to znamená, že má veľkosť $(n-1) \times (n-1)$. Táto matica sa použije v rov.(5) namiesto \mathbf{P}_1 spolu s vektorom \mathbf{x} , ktorý obsahuje teraz $(n-1)$ prvkov druhého stĺpca matice $\mathbf{U}^{(1)}$, t.j.

$$\begin{bmatrix} k \\ 0 \\ \dots \\ 0 \end{bmatrix} = \mathbf{P}'_2 \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_{n-1} \end{bmatrix} \quad (7)$$

Nakoniec maticu \mathbf{P}'_2 prispôbíme pomocou jednotkovej matici tak, aby sme získali maticu \mathbf{P}_2 požadovaného rozmeru $n \times n$

$$\mathbf{P}_2 = \begin{bmatrix} \mathbf{E} & 0 \\ 0 & \mathbf{P}'_2 \end{bmatrix} \quad (8)$$

Jej súčinom s maticou $\mathbf{U}^{(1)}$ získame maticu $\mathbf{U}^{(2)}$, ktorej teraz už prvé dva riadky sú identické s riadkami hľadanej matice \mathbf{R}

$$\mathbf{U}^{(2)} = \mathbf{P}_2 \mathbf{U}^{(1)} \quad (9)$$

Po $(n-1)$ krokoch dostaneme maticu $\mathbf{U}^{(n-1)}$, ktorá je identická s požadovanou maticou \mathbf{R}

$$\mathbf{P}_{n-1}\mathbf{P}_{n-2}\cdots\mathbf{P}_1\mathbf{U} = \mathbf{R} \quad (10)$$

Všetky matice \mathbf{P}_k sú ortogonálne, preto súčin matíc \mathbf{P} je tiež ortogonálny, pričom

$$\mathbf{P} = \mathbf{P}_{n-1}\mathbf{P}_{n-2}\cdots\mathbf{P}_1 \text{ a tiež platí } \mathbf{P}^T\mathbf{P} = \mathbf{E} \quad (11)$$

Pre maticu \mathbf{U} môžeme napísať, že $\mathbf{U} = \mathbf{P}^T\mathbf{R}$. Ďalej musíme vybrať taký vektor \mathbf{w} , ktorý vyhovuje rovnici $k\mathbf{e}_1 = \mathbf{P}'_k\mathbf{x}$. Tento vektor je daný vzťahom [4]

$$\mathbf{w} = \frac{\mathbf{x} - k\mathbf{e}_1}{\|\mathbf{x} - k\mathbf{e}_1\|} \quad (12)$$

Householderov algoritmus ortogonalizácie vypočíta všetky ortogonálne bázové funkcie naraz, kým Gram-Schmidtov algoritmus [14] počíta s postupnou ortogonalizáciou jednoduchých bázových funkcií. Householderov algoritmus je matematicky lepšie predložený v tom, že všetky výpočty sú uskutočnené prostredníctvom ortogonálnych matíc. Na druhej strane, ak lineárne nezávislé funkcie vstupujú do Schmidtového algoritmu, môže sa stať, že výsledné funkcie nie sú ortogonálne, čo je spôsobené akumuláciou chýb v procese zaokrúhľovania [16]. Hoci oba algoritmy sú celkom rozdielne, dávajú tie isté ortogonálne funkcie, ak zanedbáme vplyv zaokrúhľovacích chýb a škálovacieho faktora. Householderov algoritmus dáva ortogonálne bázové funkcie v riadkoch na rozdiel od Schmidtovho algoritmu, ktorý dáva ortogonálne bázové funkcie po stĺpcoch, pričom bázové funkcie u Householderovho algoritmu sú dokonca ortonormálne. Pri Schmidtovej metóde sa ortogonálne bázové funkcie musia ešte normalizovať [14], [15].

3. Simulačné výsledky

Ortogonalizáciu zovšeobecnenej DCT vzhľadom na ľubovoľný tvar segmentu sme vykonali pomocou Householderovho algoritmu. Segmenty sa odlišovali svojim tvarom a počtom op, ktoré obsahovali.

3.1 Pre syntetizovaný segment

Uvažujme segment zvoleného tvaru reprezentovaný maticou \mathbf{X} o rozmere 4×4

$$\mathbf{X} = \begin{pmatrix} 0 & 10 & 12 & 0 \\ 6 & 18 & 20 & 8 \\ 3 & 12 & 4 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix} \quad (13)$$

V ďalšom si názorne ukážeme, ako sa budú z ortogonálnej bázy 2-rozmernej DCT vyberať funkcie podľa daného tvaru hore uvedeného segmentu, ktoré sa potom budú ortogonalizovať Householderovým algoritmom. Z definičných vzťahov pre DCT vieme získať úplný systém 2-rozmerných diskretných ortogonálnych kosínusových funkcií, pričom vyjdeme z jadra 1-rozmernej DCT o veľkosti $N = 4$. To je uvedené v rov. (14), kde vo výslednej matici je v každom jej riadku možné vidieť hodnoty 1-rozmerných diskretných kosínusových funkcií.

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 0,707 & 0,707 & 0,707 & 0,707 \\ 0,924 & 0,383 & -0,383 & -0,924 \\ 0,707 & -0,707 & -0,707 & 0,707 \\ 0,383 & -0,924 & 0,924 & -0,383 \end{bmatrix} = \begin{bmatrix} 0,5 & 0,5 & 0,5 & 0,5 \\ 0,653 & 0,271 & -0,271 & -0,653 \\ 0,5 & -0,5 & -0,5 & 0,5 \\ 0,271 & -0,653 & 0,653 & -0,271 \end{bmatrix} \quad (14)$$

Zo systému 1-rozmerných diskretných kosínusových funkcií získame pomocou ich vzájomných súčinov systém 2-rozmerných diskretných kosínusových funkcií veľkosti 4×4 tvoriacich jadro 2-rozmernej DCT. Na obr.4. je kvôli veľkosti zobrazená iba časť tohto 2-rozmerného systému ortogonálnych bázových funkcií DCT (úplný systém by v tomto prípade tvorilo 16 týchto 2-rozmerných funkcií). V prvom riadku a v prvom stĺpci sú uvedené prvé dve 1-rozmerné diskretné kosínusové funkcie idúce za sebou (t.j funkcie z prvých dvoch riadkov matice z rov.14). V každom štvorci je tak možné vidieť 2-rozmerné diskretné kosínusové funkcie, ktoré vznikli prenasobením hodnôt z prvého riadku a prvého stĺpca. Žltou farbou je vyznačený tvar segmentu (rov.13) v prvej 2-rozmernej bázovej funkcii (submatici) DCT a podľa tohto tvaru budeme vyberať iba tie hodnoty z bázových funkcií, ktoré sú na tých istých pozíciách ako op segmentu a ukladať ich do vektorov \mathbf{x} .

0	0,5	0,5	0,5	0,5	0,653	0,271	-0,271	-0,653
0,5	0,25	0,25	0,25	0,25	0,327	0,136	-0,136	-0,327
0,5	0,25	0,25	0,25	0,25	0,327	0,136	-0,136	-0,327
0,5	0,25	0,25	0,25	0,25	0,327	0,136	-0,136	-0,327
0,5	0,25	0,25	0,25	0,25	0,327	0,136	-0,136	-0,327
0,653	0,327	0,327	0,327	0,327	0,426	0,177	-0,177	-0,426
0,271	0,136	0,136	0,136	0,136	0,177	0,073	-0,073	-0,177
-0,271	-0,136	-0,136	-0,136	-0,136	-0,177	-0,073	0,073	0,177
-0,653	-0,327	-0,327	-0,327	-0,327	-0,426	-0,177	0,177	0,426
.
.
.
.

Obr.4. Časť bázy 2-rozmerných ortogonálnych kosínusových funkcií s vyznačeným tvarom daného segmentu.

Daný segment má 10 prvkov, preto vyberieme iba 10 hľadaných funkcií z 10-tich submatic 2-rozmernej DCT z celkového počtu 16. Načítavame po riadkoch, zhora dole a ukladáme do vektorov. Takto sme získali bázu 10 kosínusových funkcií (vektorov) podľa tvaru segmentu ($\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_9$), ktorá však nie je ortogonálna. Na jej ortogonalizáciu použijeme Householderov algoritmus uvedený v predchádzajúcej kapitole.

$$\begin{array}{l}
 \mathbf{x}_0 = \begin{pmatrix} 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \end{pmatrix} \quad \mathbf{x}_1 = \begin{pmatrix} 0,136 \\ -0,136 \\ 0,327 \\ 0,136 \\ -0,136 \\ -0,327 \\ 0,327 \\ 0,136 \\ -0,136 \\ 0,136 \end{pmatrix} \quad \mathbf{x}_2 = \begin{pmatrix} -0,25 \\ -0,25 \\ 0,25 \\ -0,25 \\ 0,25 \\ 0,25 \\ -0,25 \\ -0,25 \\ -0,25 \\ -0,25 \end{pmatrix} \quad \mathbf{x}_3 = \begin{pmatrix} -0,327 \\ 0,327 \\ 0,136 \\ -0,327 \\ 0,327 \\ -0,136 \\ 0,136 \\ -0,327 \\ 0,327 \\ -0,327 \end{pmatrix} \quad \mathbf{x}_4 = \begin{pmatrix} 0,327 \\ 0,327 \\ 0,136 \\ 0,136 \\ 0,136 \\ 0,136 \\ -0,136 \\ -0,136 \\ -0,136 \\ -0,327 \end{pmatrix} \\
 \\
 \mathbf{x}_5 = \begin{pmatrix} 0,177 \\ -0,177 \\ 0,177 \\ 0,073 \\ -0,073 \\ -0,177 \\ -0,177 \\ -0,073 \\ -0,177 \\ 0,073 \\ -0,177 \end{pmatrix} \quad \mathbf{x}_6 = \begin{pmatrix} -0,327 \\ -0,327 \\ 0,136 \\ -0,136 \\ -0,136 \\ 0,136 \\ -0,136 \\ 0,136 \\ -0,136 \\ 0,136 \\ 0,327 \end{pmatrix} \quad \mathbf{x}_7 = \begin{pmatrix} -0,426 \\ 0,426 \\ 0,073 \\ -0,177 \\ 0,177 \\ -0,073 \\ -0,073 \\ 0,177 \\ -0,177 \\ 0,426 \end{pmatrix} \quad \mathbf{x}_8 = \begin{pmatrix} 0,25 \\ 0,25 \\ -0,25 \\ -0,25 \\ -0,25 \\ -0,25 \\ -0,25 \\ -0,25 \\ -0,25 \\ 0,25 \end{pmatrix} \quad \mathbf{x}_9 = \begin{pmatrix} 0,136 \\ -0,136 \\ -0,327 \\ -0,136 \\ 0,136 \\ 0,327 \\ -0,327 \\ -0,136 \\ 0,136 \\ 0,136 \end{pmatrix}
 \end{array}$$

Po ortogonalizácii Householderovym algoritmom dostaneme maticu \mathbf{P} ortogonálnych kosínusových bázových funkcií (ortogonálne bázové funkcie sú v riadkoch matice \mathbf{P} a po jej transponovaní dostaneme ortogonálne bázové funkcie v stĺpcoch). Platí

$$\mathbf{P} = \{\mathbf{p}_0; \mathbf{p}_1; \dots; \mathbf{p}_n\}, \quad n = 0, 1, \dots, (M_1 \times M_2) - 1 \quad (15)$$

t.j. (10×1)-1, pričom

$$\begin{array}{l}
 \mathbf{p}_0 = \begin{pmatrix} 0,3162 \\ 0,3162 \\ 0,3162 \\ 0,3162 \\ 0,3162 \\ 0,3162 \\ 0,3162 \\ 0,3162 \\ 0,3162 \\ 0,3162 \end{pmatrix} \quad \mathbf{p}_1 = \begin{pmatrix} 0,1364 \\ -0,2778 \\ 0,4292 \\ 0,1364 \\ -0,2778 \\ -0,5706 \\ 0,4292 \\ 0,1364 \\ -0,2778 \\ 0,1364 \end{pmatrix} \quad \mathbf{p}_2 = \begin{pmatrix} -0,2389 \\ -0,1549 \\ 0,4059 \\ -0,2389 \\ -0,1549 \\ 0,6085 \\ 0,4059 \\ -0,2389 \\ -0,1549 \\ -0,2389 \end{pmatrix} \quad \mathbf{p}_3 = \begin{pmatrix} -0,2727 \\ 0,3635 \\ 0,2259 \\ -0,2727 \\ 0,3635 \\ -0,4518 \\ 0,2259 \\ -0,2727 \\ 0,3635 \\ -0,2727 \end{pmatrix} \quad \mathbf{p}_4 = \begin{pmatrix} 0,5201 \\ 0,3468 \\ 0,2155 \\ 0,2155 \\ 0,0421 \\ 0,0000 \\ -0,2155 \\ -0,2155 \\ -0,3888 \\ -0,5201 \end{pmatrix} \\
 \\
 \mathbf{p}_5 = \begin{pmatrix} 0,2053 \\ -0,5225 \\ 0,3747 \\ 0,0850 \\ -0,0634 \\ 0,0000 \\ -0,3747 \\ -0,0850 \\ 0,5859 \\ -0,2053 \end{pmatrix} \quad \mathbf{p}_6 = \begin{pmatrix} -0,3377 \\ 0,2164 \\ 0,5596 \\ -0,1399 \\ 0,0263 \\ 0,0000 \\ -0,5596 \\ 0,1399 \\ -0,2427 \\ 0,3377 \end{pmatrix} \quad \mathbf{p}_7 = \begin{pmatrix} -0,4432 \\ -0,2773 \\ 0,0000 \\ 0,6625 \\ 0,4734 \\ 0,0000 \\ 0,0000 \\ -0,0861 \\ -0,1961 \\ -0,1332 \end{pmatrix} \quad \mathbf{p}_8 = \begin{pmatrix} 0,1724 \\ -0,0145 \\ 0,0000 \\ 0,0930 \\ 0,0247 \\ 0,0000 \\ 0,0000 \\ -0,8129 \\ -0,0102 \\ 0,5476 \end{pmatrix} \quad \mathbf{p}_9 = \begin{pmatrix} 0,3115 \\ -0,3866 \\ 0,0000 \\ -0,4787 \\ 0,6599 \\ 0,0000 \\ 0,0000 \\ 0,0922 \\ -0,2733 \\ 0,0750 \end{pmatrix} \quad (16)
 \end{array}$$

Získané ortogonálne bázové funkcie musia spĺňať podmienky ortogonalnosti a ortonormality, t.j. všeobecne

$$\sum_{k=0}^{M-1} \sum_{m=0}^{M-1} \mathbf{p}_k \mathbf{p}_m = 0, \text{ pre } k \neq m$$

a

$$\sum_{k=0}^{M-1} \sum_{m=0}^{M-1} \mathbf{P}_k \mathbf{P}_m = 1, \text{ pre } k = m \quad (17)$$

pričom v Tab.1 sú uvedené dosiahnuté výsledky testu ortonormality pre hore uvedené bázové funkcie.

Tab.1 Test ortogonality a ortonormality nájdených bázových funkcií Householderovým algoritmom

Test ortogonality:			Test ortonormality:
p0.p1= 9,9747e-018	p1.p8= -5,1046e-017	p4.p5= -2,2437e-016	p0.p0=1,0000
p0.p2= -4,2962e-018	p1.p9= 2,2128e-016	p4.p6= 1,9472e-016	p1.p1=1,0000
p0.p3= -4,6289e-017	p2.p3= -2,4239e-016	p4.p7= -4,1110e-016	p2.p2=1,0000
p0.p4= 8,2250e-017	p2.p4= -3,8171e-016	p4.p8= 9,0721e-017	p3.p3=1,0000
p0.p5= 5,8635e-017	p2.p5= -2,9860e-017	p4.p9= -2,6484e-016	p4.p4=1,0000
p0.p6= -1,2235e-016	p2.p6= -1,2842e-016	p5.p6= -1,5898e-016	p5.p5=1,0000
p0.p7= -7,4569e-017	p2.p7= 2,5022e-016	p5.p7= 1,2795e-016	p6.p6=1,0000
p0.p8= 2,4855e-017	p2.p8= -5,3505e-017	p5.p8= -3,9241e-017	p7.p7=1,0000
p0.p9= 5,2079e-017	p2.p9= 3,7176e-017	p5.p9= -1,5213e-016	p8.p8=1,0000
p1.p2= 2,0961e-016	p3.p4= 1,4929e-016	p6.p7= -1,7622e-016	p9.p9=1,0000
p1.p3= -1,8924e-016	p3.p5= -1,0947e-016	p6.p8= 8,3416e-017	
p1.p4= -6,0826e-016	p3.p6= -4,3104e-017	p6.p9= -2,7342e-017	
p1.p5= 3,5927e-016	p3.p7= 2,1125e-017	p7.p8= -1,5884e-016	
p1.p6= -1,0032e-016	p3.p8= 4,1335e-018	p7.p9= -1,0018e-016	
p1.p7= 5,0746e-016	p3.p9= -1,7499e-016	p8.p9= -1,0297e-016	

Vlastnosť ortogonality bázových funkcií dáva možnosť použiť ich pri výpočte priamej a spätnej zovšeobecnenej DCT pre daný vstupný segment (matica \mathbf{X}) podľa nasledujúcich transformačných vzťahov (vo vektorovom tvare)

$$\bar{\mathbf{Y}} = \mathbf{P}\bar{\mathbf{X}}, \text{ priama DCT}, \quad (18)$$

$$\bar{\mathbf{X}}_s = \mathbf{P}^T \bar{\mathbf{Y}}, \text{ priama DCT}, \quad (19)$$

kde \mathbf{P} je transformačná ortogonálna matica t.j. platí $\mathbf{P}^T \mathbf{P} = \mathbf{E}$

$$\mathbf{P} = \{\mathbf{p}_0; \mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3; \mathbf{p}_4; \mathbf{p}_5; \mathbf{p}_6; \mathbf{p}_7; \mathbf{p}_8; \mathbf{p}_9\} \quad (20)$$

$$\bar{\mathbf{X}} = [10, 12, 6, 18, 20, 8, 3, 12, 4, 2] \quad (21)$$

Matica spektrálnych koeficientov

$$\mathbf{Y} = \begin{pmatrix} 0 & 30,0416 & -4,9742 & 0 \\ -7,0890 & 0,0548 & 9,5475 & -1,9177 \\ 0,2888 & 11,5489 & -4,9836 & 0 \\ 0 & 3,2201 & 0 & 0 \end{pmatrix} \quad (22)$$

a matica rekonštruovaného obrazu

$$\mathbf{X}_s = \begin{pmatrix} 0 & 10 & 12 & 0 \\ 6 & 18 & 20 & 8 \\ 3 & 12 & 4 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix} \quad (23)$$

Stredná kvadratická odchýlka dosiahla hodnotu

$$\xi = 1/M \sum_{n=0}^{M-1} (\overline{X}(nT) - \overline{X_s}(nT))^2 = 5,5861e^{-029} \quad (24)$$

, kde $M=10$. Z výslednej hodnoty strednej kvadratickej chyby je zrejmé, že vstupný segment a rekonštruovaný segment môžeme považovať za totožný, čo znamená, že vypočítané ortogonálne bázové funkcie Householderovým algoritmom sú správne vypočítané.

3.2 Pre reálne obrazové segmenty

Navrhnutý program na simuláciu zovšeobecnenej DCT umožňuje načítať jednu alebo viacero šedotónových snímok, ďalej ortogonalizáciu je možné realizovať dvoma spôsobmi: 1) s určením okna segmentu a 2) s vyhľadáním hraničných blokov, pričom sa ortogonalizácia vykonáva vždy na aktívnej snímke. Pri 1) type ortogonalizácie program čaká na výber okna zo segmentu na aktívnej snímke a po zadaní veľkosti okna (môže byť veľkosti 4×4 , 8×8 , 16×16) nastáva proces ortogonalizácie Householderovym algoritmom. Pri 2) type ortogonalizácie sa najskôr nastavuje veľkosť hraničných blokov a potom je potrebné nastaviť ďalší parameter a to transparentnosť, čo znamená udať počet nulových op v hraničnom bloku.

Po nastavení transparentnosti si môžeme vybrať, či budeme robiť ortogonalizáciu hraničných blokov alebo selekciu. Po zvolení „ortogonalizácie“ sa vykoná ortogonalizácia Householderovym algoritmom pre nastavené hraničné bloky a po skončení procesu môže nasledovať transformácia alebo filtrácia, ako v predchádzajúcom bode. Po zvolení „selekcie“ sa na obrazovke zobrazí dialóg, v ktorom si vyberieme, či budeme zobrazovať hraničné bloky alebo vnútorné bloky segmentu. Položka v programe „transformácia“ je aktuálna až po skončení ortogonalizácie. „Filtrácia“ je aktuálna po priamej transformácii, pričom si môžeme zvoliť buď prahovú alebo zonálnu filtráciu [7].

Pre ilustráciu uvádzame dosiahnuté výsledky pre reprezentačný segment (obr.5 vľavo). Pre tento reálny segment sme robili ortogonalizáciu pomocou Householderovho algoritmu. Počet obrazových prvkov tohto segmentu je 115 a veľkosť okna, v ktorom sa daný segment nachádza je 16×32 . Po vykonaní ortogonalizácie sme transformovali tento reálny segment priamou zovšeobecnenou DCT (obr.5 v strede). Správnosť výpočtu Householderovým algoritmom sme overili spätnou transformáciou (obr.5 vpravo). Segment po spätnej zovšeobecnenej DCT je rovnaký ako originálny segment vstupujúci do výpočtu, čo sa potvrdilo aj dosiahnutou nulovou strednou kvadratickou chybou rekonštruovaného segmentu.



Obr.5 Vľavo reálny segment, v strede segment po priamej zovšeobecnenej DCT s vykonanou Householderovou ortogonalizáciou, vpravo segment po spätnej zovšeobecnenej DCT.

Niekedy je však potrebné ortogonalizovať iba hraničné bloky segmentu napr. vtedy, ak máme veľký segment (napr. snímka typu hlava-plecia), kde sa netransparentné bloky vo vnútri segmentu transformujú pomocou klasickej DCT [14] a polotransparentné bloky sa ortogonalizujú pomocou uvedeného Householderovho algoritmu.

4. Záver

V článku bola vykonaná zovšeobecnená diskretná kosínusová transformácia vzhľadom na ľubovoľný tvar segmentu s použitím kosínusových básových funkcií. Ortogonalizácia daných básových funkcií pre zvolený tvar segmentu bola uskutočnená pomocou Householderovho algoritmu. Navrhnutý simulačný program umožňuje realizovať zovšeobecnenú DCT pre zadané vstupné segmenty. Householderov algoritmus ortogonalizácie je veľmi presný, stredná kvadratická chyba vychádzala pre jednotlivé ľubovoľné testované obrazové segmenty nulová, rádovo na 10-19. Táto metóda sa javí vhodnejšia na spracovanie vizuálneho objektu a pri kompresii údajov vykazuje menšiu chybu pri spätnej transformácii v porovnaní so Schmidtovym algoritmom ortogonalizácie. Jej nevýhoda spočíva v dlhšom trvaní výpočtu ortogonalizácie a z toho vyplývajúce vysoké nároky na procesor a pamäť počítača. Získané poznatky je možné použiť v oblasti transformačného kódovania obrazov, ktoré používa oblastne-orientovanú obrazovú segmentáciu.

Literatúra

1. Mihalík, J., Štandardný videokodek MPEG-4. Slaboproudý obzor, 2003, roč. 60, č.2, s.7-11.
2. Mihalík, J., Štandardný videokodek H.263 a jeho porovnanie s H.261. Slaboproudý obzor, 2003, roč. 60, č.1, s.1-7.
3. MPEG Requirements Group. MPEG-4 requirements. ISO/IEC JTC1/SC29/WG11 N1595, Sevilla MPEG meeting, February 1997
4. Gilge, M. - Engelhardt, T.- Mehlan, R., Coding of arbitrarily shaped image segments based on a generalized orthogonal transform. Signal Processing: Image Communication.,vol 1, no.2, 1989, pp.153-180.
5. Gilge, M., Region-oriented transform coding (ROTC) of images. ICASSP, Conference: Acoustics, Speech, and Signal Processing, 1990.
6. Correia, P. - Pereira, F., Video Analysis for Coding: Objectives, Features and Methods, April 1997.
7. Mihalík, J., Kódovanie obrazu vo videokomunikáciách. TU-FEI Košice, 2001.
8. Mihalík, J., Štandardný videokodek H.264/AVC, Slaboproudý obzor, roč. 67, č.4, 2011, str.4-10.
9. Fauzi, M.F.A. - Lewis, P.H., Block-based Against Segmentation-based Texture Image Retrieval. Journal of Universal Computer Science, vol. 16, no. 3 (2010), 402-423.
10. Seyedin, S.A. - Fauzi, M.F.A. - Anuar, F.M., 1-D DCT Shape Faecture for Image Retrieval. IEEE Int. Conf. on Signal and Image Processing Applications, 2009.
11. Březina, A. - Polec, J. - Pavlovičová, J. - Kolingerová, I. - Bandzi, P., Region-based texture coding at very low bitrates. Journal of Electrical Engineering, vol. 56, no. 1-2, 2005, 36-40.
12. Kaup, A. - Aach, T., Coding of Segmented Images Using Shape-Independent Basis Functions, IEEE Transactions on Image Processing, no. 7 (1998), 937-947.
13. Eden, M. - Kocher, M., On the Performance of a Contour Coding Algorithm in the

- Context of Image Coding. Part I: Contour Segment Coding, Signal Processing 8 (1995), 381-386.
14. Gladišová, I. - Mihalík, J., DCT vizuálneho objektu pomocou ortogonalizácie jej básových funkcií v oblastiach hraničných blokov. Electrical Engineering and Informatics 3, Proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, 2012, s. 310-314.
 15. Persson P., Lecture 5: Gram-Schmidt Orthogonalization. [Online]: <http://persson.berkeley.edu/18.335/lec5handout6pp.pdf>
 16. Giraud, L. - Langou, J. - Rozlonik, M., The Loss of Orthogonality in the Gram-Schmidt Orthogonalization Process. Int. Journal of Computers and Mathematics with Applications, Vol.50, 2005, pp.1069-1075.
 17. Gladišová, I., Mihalík, J., Zavacký, J., Obrysová reprezentácia a kódovanie binárneho tvaru videoobjektu. Slaboproudý obzor, 2010, roč. 66, č.3, s.14-18.
 18. Liu, Jin-Zuo - Ding, Jian-Jiun, Image Compression, [Online]: <http://disp.ee.ntu.edu.tw/meeting/□□/imagecompression.ppt>
 19. Gladišová, I. - Mihalík, J. - Petráš, J., Vnútrošnímková segmentácia obrazu. In: Poster.us.sk., portál pre odborné publikovanie, roč. 8, č. 2 (2015), s. 1-18.

Spoluautormi článku sú Ján Mihalík a Miroslav Tóth, Laboratórium číslicového spracovania obrazov a videokomunikácií, Katedra elektroniky a multimediálnych telekomunikácií, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach.
