

فشرده سازی لایه ای اندیسهای TSVQ

و کاربرد آن در اینترنت

سید حمید غفوری^۱، جمشید شنبه زاده^۲، حمید سلطانیان زاده^۳

^۱ دانشگاه آزاد اسلامی، واحد نجف آباد، دانشکده تحصیلات تکمیلی گروه برق و کامپیوتر

^۲ دانشگاه تربیت معلم، گروه مهندسی کامپیوتر

^۳ دانشگاه تهران، دانشکده فنی، گروه برق و کامپیوتر

shghalfuri@yahoo.com, ^۲ jamshid@saba.tmu.ac.ir, ^۳ hszadeh@ut.ac.ir

چکیده:

در این مقاله یک روش جدید برای انتقال و دریافت تصویر به صورت لایه ای مناسب برای اینترنت ارائه می شود. در فشرده سازی تصویر به روش TSVQ، نمایه های تصویر دارای ساختار سلسله مراتبی هستند این ساختار علاوه بر اطلاعات همبستگی درون بلوکی، اطلاعات همبستگی بین بلوکی تصویر را نیز در بر دارد. در روش جدید با استفاده از این ویژگی لایه های نمایه ها را با یک روش بدون تلف فشرده کرده ایم. در هنگام بازسازی نیز تصویر به صورت لایه به لایه بازسازی می شود این خصوصیت باعث شده تا این روش برای انتقال تصاویر از طریق اینترنت مناسب باشد. آزمایشات انجام شده بر روی ۲۶ تصویر نشان می دهد که نرخ فشرده سازی این روش برای ۴ بیت اول نمایه ها ۱/۵۵ برابر نسبت به روش TSVQ افزایش یافته است.

کلمات کلیدی: بازبازی تصاویر، فشرده سازی تصاویر، اینترنت، TSVQ.

یکی از مشکلات انتقال اطلاعات از طریق اینترنت ارسال تصاویر است. تصاویر به دلیل حجم بالای اطلاعات، به زمان زیادی جهت انتقال نیاز دارد، بنابراین باید به گونه ای مناسب آنها را فشرده کرد. اعوجاج سیگنال؛ نرخ بیت و پیچیدگی کد کردن^۱، سه پارامتر مهمی هستند که اغلب روشهای فشرده سازی سعی در بهینه کردن آنها دارند. اما در کاربردهای اینترنتی روش دریافت تصاویر نیز می تواند به عنوان یک فاکتور مهم در نظر گرفته شود.

در بسیاری از مراجعات به صفحات اینترنتی، تصویر بار شده، تصویر مورد نظر کاربر نیست. اگر روش فشرده سازی به گونه ای باشد که در هنگام دریافت، تصویر به صورت لایه به لایه بازسازی شود. کاربر پس از تکمیل شدن چند لایه از تصویر می تواند تشخیص دهد که آیا تصویر بار شده، تصویر مورد نظر او هست یا خیر و در صورتیکه تصویر مطلوب او نباشد به صفحه دیگری برود. بدین ترتیب در وقت کاربران صرفه جویی زیادی خواهد شد.

روش Tree Structured Vector Quantization (TSVQ) یکی از روشهای مناسب برای این کاربرد است. این روش علاوه بر داشتن یک کد کننده-کد گشای^۲ ساده، در کتاب کد خود از یک ساختار درختی استفاده می کند، که این قابلیت را دارد تا تصویر را به صورت لایه به لایه ارسال و دریافت کند. اما از آنجا که این روش فقط از همبستگی درون بلوکی برای فشرده کردن تصویر استفاده می کند، دارای نرخ فشرده سازی بالایی نیست.

تا کنون روشهای متعدد فشرده سازی مبتنی بر Vector Quantization (VQ) ارائه شده است. که در آنها همبستگی بین بلوکی^۳ نیز در نظر گرفته شده است [۱، ۲، ۳، ۴، ۵، ۶، ۷]. یک گروه از روشهای سریع و ساده، VQ های مبتنی بر فشرده سازی نمایه^۴ است که توسط شنبه زاده و همکاران [۵، ۶، ۷] ارائه شده است. این روشها همانند روش اصلی VQ از مزیت سادگی کد کننده - کدگشا برخوردار هستند و با به کارگیری یک روش فشرده سازی بی تلف بر روی نمایه های بدست آمده از روش اصلی VQ نرخ فشرده سازی را به طور قابل ملاحظه ای افزایش داده اند. این روشها از همبستگی بین بلوکی برای فشرده سازی نمایه ها استفاده می کنند [۷، ۸].

ما در این مقاله با ارائه یک روش فشرده سازی بدون تلف، مبتنی بر رابطه نیاکانی نمایه های TSVQ، لایه های نمایه های TSVQ را کد کرده ایم و نرخ فشرده سازی را به طور قابل ملاحظه ای افزایش داده ایم. بنا بر این نوآوری این مقاله از دو نگرش قابل بررسی است. نگرش اول انتخاب یک روش فشرده سازی مناسب برای اینترنت، که قابلیت دریافت تصاویر به صورت لایه ای را داشته باشد. و نگرش دوم ارائه یک الگوریتم جدید برای بالاتر بردن نرخ فشرده سازی این روش. در این مقاله پس از بخش اول که به مقدمه اختصاص داده شد مطالب ذیل ارائه می شود. در بخش دوم الگوریتم روش جدید و نتایج تئوری آن بیان می شود. در بخش سوم بحث و بررسی نتایج بدست آمده از شبیه سازیها مطرح می شود. بخش پایانی به نتیجه گیری اختصاص داده شده است.

^۱ - Coding

^۲ load

^۳ codec

^۴ Inter-Block Correlation

^۵ Index Compressed VQ

۲- الگوریتم فشرده سازی :

روش فشرده سازی TSVQ از یک مجموعه کتاب کد با ساختار درختی با n سطح استفاده می کند ، که اندازه کتاب کد سطح k ام 2^k ($1 \leq k \leq n$) و نمایه های آن k بیتی است . در این روش تصویر اولیه به بلوک هایی تقسیم و برای هر بلوک نمایه ای از کتاب کد ایجاد می شود . به دلیل وجود همبستگی میان اجزاء تصویر ممکن است دو بلوک مجاور دارای نمایه های یکسان باشند و یا در $1, 2, \dots, k-1$ بیت کم ارزش از نمایه هایشان اختلاف داشته باشند. در نتیجه احتمال یکسان بودن بیتهای با ارزش نمایه های دو بلوک مجاور بسیار بالاست . در روش جدید با استفاده از این ویژگی ، نمایه های TSVQ به صورت بدون تلف فشرده شده اند.

در این الگوریتم بیتهای نمایه ها به صورت لایه به لایه فرستاده می شود . از آنجا که احتمال یکسان بودن این بیتها بالاست ، در هر مرحله رشته هایی از صفر و یک ها باید ارسال گردد . این رشته ها را می توان با روش Run-length (RL) کد کرد . در RL به جای چند صفر یا یک متوالی ، یک عدد که نشان دهنده تعداد آنهاست فرستاده می شود [10] . در حقیقت بیتهای اول نمایه ها تصویر را به دو گونه ناحیه تقسیم می کند . ناحیه صفرها و ناحیه یک ها. بیتهای سطح دوم نمایه ها هر کدام از این ناحیه ها را به دو گونه ناحیه جدید و به همین ترتیب بیت های سطح k ام نمایه های هر کدام از نواحی تشکیل شده از بیت های سطح $(k-1)$ ام را به دو گونه ناحیه جدید تقسیم می کند .

در سطوح بالایی رشته صفرها و یک ها طولانی ترند و با اعمال RL نرخ فشرده سازی بالایی به دست خواهد آمد . اما هرچه به طرف سطوح پایین تر نمایه های TSVQ پیش برویم ، اندازه این ناحیه ها و در نتیجه حداکثر طول رشته ها کوچکتر و در نتیجه نرخ فشرده سازی کمتر خواهد شد . تا جاییکه دیگر کد کردن آنها به روش RI منجر به فشرده سازی بیشتر نخواهد شد . از آن سطح به بعد بیتها را به همان گونه که هست - بدون کد کردن- ارسال می کنیم. در این مقاله با انجام شبیه سازیهایی نشان داده ایم که تاچه سطحی را باید کد کرد و نرخ فشرده سازی حاصل از این روش را نیز بدست آورده ایم . برای بدست آوردن نرخ فشرده سازی این روش باید نرخ فشرده سازی روش RL برای هر سطح را حساب کرد که برابر است با [10] .

$$\epsilon_{\max} = \frac{\bar{r}_0 + \bar{r}_1}{\bar{H}_0 + \bar{H}_1} \quad (1)$$

که \bar{r}_0 و \bar{r}_1 به ترتیب عبارتند از متوسط طول رشته های صفر و یک و \bar{H}_0 و \bar{H}_1 نیز Entropy آنها هستند. ϵ_{\max} نیز نشان دهنده نرخ فشرده سازی می باشد. مقادیر \bar{r}_1 و \bar{H}_1 نیز به ترتیب از فرمولهای زیر حساب می شود: [9]

$$\bar{r}_1 = \sum_{r=1}^m r \times p(r) \quad (2)$$

$$\bar{H}_1 = \sum_{r=1}^m p(r) \times \log_2(p(r)) \quad (3)$$

که r و $P(r)$ به ترتیب ، طول رشته ای از یک ها ، احتمال وقوع رشته ای به طول r و حداکثر طول یک رشته از یک هاست . چون احتمال وقوع صفر و یک برابر است در نتیجه $\bar{r}_0 = \bar{r}_1$ و $\bar{H}_0 = \bar{H}_1$ است ، بنابراین برای به دست آوردن مقدار ϵ_{\max} کفایت تا مقادیر \bar{r}_1 و \bar{H}_1 را به دست آوریم.

در فرمول بالا مقدار m برای بیتهای سطح اول برابر با حداکثر نمایه های درون یک سطر از تصویر است ، که بستگی به دقت تصویر دارد ، و مقدار آن برای سطح k برابر با مقدار \bar{r}_1 سطح $(k-1)$ ام است .

همچنین مقدار $p(r)$ برابر است با :

$$P(r) = \begin{cases} 0.5 \times PA_n^{r-1} \times (1 - PA_n) & 1 \leq r < m \\ 0.5 \times PA_n^{m-1} & r = m \end{cases} \quad (4)$$

که در آن 0.5 احتمال اینکه مقدار اولین بیت سطح n ام برابر یا یک باشد و PA_n احتمال اینکه بیت سطح n ام نمایه یک بلوک با بیت سطح n ام نمایه بلوک قبلی یکسان باشد به شرط اینکه $(n-1)$ بیت قبلی آنان برابر باشد است. مقدار PA_n نیز از فرمول زیر به دست می آید:

$$PA_n = \frac{P_n \cap P_{(n-1)}}{P_{(n-1)}} \quad (5)$$

که P_i احتمال برابر بودن i بیت با ارزش نمایه یک بلوک با بلوک قبلی خودش می باشد. و چون P^n زیر مجموعه $P^{(n-1)}$ است پس داریم:

$$PA_n = \frac{P_n}{P_{(n-1)}}$$

اگر مقادیر (4) را در (2) و (3) جایگذاری کنیم خواهیم داشت .

$$\bar{r} = \bar{r}_1 = \sum_{r=1}^{m-1} r \times 0.5 \times PA_n^{r-1} \times (1 - PA_n) + 0.5 \times PA_n^{m-1} \quad (6)$$

$$\bar{H}_0 = \bar{H}_1 = \sum_{r=1}^{m-1} (0.5 \times PA_n^{r-1} \times (1 - PA_n) \times \log_2(0.5 \times PA_n^{r-1} \times (1 - PA_n)) + 0.5 \times PA_n^{m-1} \times \log_2(0.5 \times PA_n^{m-1})) \quad (7)$$

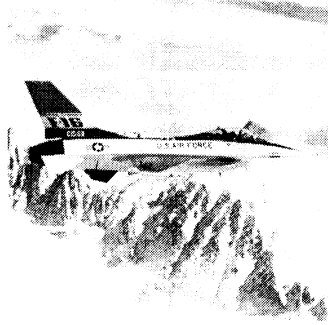
با استفاده از فرمولهای بالا و داشتن مقادیر P_n برای $(1 \leq n \leq 7)$ می توان مقدار \mathcal{E}_{\max} برای هر سطح را به دست آوریم. نرخ فشرده سازی میانگینی که از فشرده کردن k لایه نمایه ها به دست می آید برابر است با:

$$\bar{\mathcal{E}}_{\max} = \frac{k}{\sum_{i=1}^k \mathcal{E}_{\max i}} \quad (8)$$

که در آن $\mathcal{E}_{\max i}$ نشان دهنده نرخ فشرده سازی سطح i ام است .

۳- نتایج شبیه سازی :

در این بخش نتایج بدست آمده از شبیه سازیها بحث و بررسی می شود. در این مقاله آزمایشات ما بر روی ۲۶ تصویر استاندارد با ابعاد 640×480 پیکسل و بلوکهای 4×4 انجام شده که دو نمونه از آنها در تصاویر 2.1 نشان داده شده اند .



2. Airplane



1. Light

چون آزمایشات بر روی تصاویر با ابعاد 640×480 و اندازه بلوکهای 4×4 انجام شده است در نتیجه حداکثر تعداد نمایه ها در هر سطر برابر با $\frac{640}{4}$ یعنی 160 خواهد بود پس مقدار m برای بینهای سطح اول برابر با 160 خواهد بود. برای بدست آوردن PA_n ابتدا مقدار P_n برای $(1 \leq n \leq 7)$ را بدست آوردیم که در جدول ۱ نشان داده شده است.

جدول ۱ - احتمال برابر بودن n بیت با ارزش نمایه یک بلوک با بلوک واقع در سمت غرب خود

	7	6	5	4	3	2	1
Light	0.5865	0.6562	0.6660	0.7020	0.7902	0.8983	0.9487
Airplane	0.4892	0.5490	0.5862	0.6678	0.7424	0.8620	0.9352

سیس با استفاده از اطلاعات جدول ۱ و فرمول (5) مقدار PA_n را برای $1 \leq n \leq 7$ بدست آورده ایم که در جدول 2 نشان داده شده است.

جدول ۲ - احتمال برابر بودن بیت n ام نمایه یک بلوک با بلوک واقع در غرب خود به شرط برابر بودن $(n-1)$ بیت قبلی

	7	6	5	4	3	2	1
Light	0.8938	0.9853	0.9486	0.8884	0.8797	0.9469	0.9487
Airplane	0.891	0.9365	0.8778	0.8996	0.8613	0.9217	0.9352

با داشتن مقدار PA_1 از جدول ۲ و مقدار m و جایگذاری آنها در فرمولهای (6) و (7) مقدار $\bar{r}_1, \bar{r}_0, \bar{H}_1, \bar{H}_0$ برای سطح اول را بدست آورده و در فرمول (1) قرار داده ایم و مقدار \mathcal{E}_{\max} برای این سطح را بدست آورده ایم. برای بدست آوردن نرخ فشرده سازی بینهای سطح دوم نمایه ها مقدار m برابر است با l^1 سطح اول و به همین ترتیب مقدار m برای بینهای سطح i ام برابر است با مقدار l^i برای سطح $(i-1)$ ام.

نتایج بدست آمده برای چهار بیت با ارزش نمایه های تصاویر مورد آزمایش در جدول (3) آورده شده است .

جدول ۳- مقادیر \bar{H}_1 ، \bar{r}_1 و ε_{\max} برای تصاویر Light و Airplane

	Fourth bit			Third bit			Second bit			First bit		
	ε_{\max}	\bar{H}_1	\bar{r}_1	ε_{\max}	\bar{H}_1	\bar{r}_1	ε_{\max}	\bar{H}_1	\bar{r}_1	ε_{\max}	\bar{H}_1	\bar{r}_1
Light	1.2550	0.7524	0.9442	1.3853	1.2035	1.6672	2.4841	1.5940	3.9597	2.9137	3.3443	9.7445
Airplane	1.2920	0.7351	0.9498	1.2508	1.0405	1.3016	1.9121	1.6002	3.0597	2.433	3.1713	7.7159

در جدول ۴ مقدار متوسط نرخ فشرده سازی هر لایه برای ۲۶ تصویر آزمایش شده نشان داده شده است .

جدول ۴- نرخ فشرده سازی متوسط برای ۲۶ تصویر مورد آزمایش

	Fourth bit	Third bit	Second bit	First bit
$\bar{\varepsilon}_{\max}$	1.1235	1.2972	1.8421	2.6331

با توجه به نتایج بدست آمده بهتر است ۴ بیت با ارزش نمایه های TSVQ را به صورت کد شده و بقیه را به همان صورتی که هست بفرستیم .

مقایسه این روش با روش TSVQ نشان می دهد که این روش بهبود قابل ملاحظه ای نسبت به روش TSVQ داشته است . نرخ فشرده سازی این روش برای ۴ بیت اول نمایه ها نسبت به TSVQ ، ۱/۵۵ برابر بهتر شده است .

۴- نتیجه گیری :

در این مقاله یک روش فشرده سازی جدید مبتنی بر TSVQ مناسب برای اینترنت ارائه شده است . که بر اساس ساختار لایه ای TSVQ هر سطح را به صورت جداگانه با اعمال RL کد میکند . این روش علاوه بر داشتن کدگشای ساده و نرخ فشرده سازی مناسب ، تصاویر را به صورت لایه ای بازسازی می کند، که این خواص باعث شده تا این روش به روشی مناسب جهت انتقال تصاویر از طریق اینترنت تبدیل شود . در این مقاله روش جدید را با روش TSVQ مقایسه کرده ایم . نتایج نشان می دهد که نرخ فشرده سازی این روش نسبت به روش TSVQ ، ۱/۵۵ برابر بهتر شده است .

1. T. Lookabugh, E.E. Riskin, P.A. Chou and R.M. Gray, "Variable Rate Vector Quantization for Speech, Image and Video Compression", IEEE Transaction on Communication, Vol.41, NO.1, Jan.1993
2. R. Aravind and A. Gersho, "Low-Rate Image Coding with Finite-State Vector Quantization", ICASSP, PP. 137-140, 1986
3. N.M. Nasrabadi and Y. Feng, "A Multilayer Address Vector Quantization Technique", IEEE Transaction on Circuit and Systems, Vol.37, NO.7, PP.912-921, July 1990
4. N.M. Nasrabadi and Y. Feng, "Image Compression Using Address-Vector Quantization", IEEE Transaction on Communication, Vol.38, PP.2166-2173, Dec.1990
5. J. Shanbehzadeh and P.O. Ogunbona, "Index Compressed Image Adaptive Vector Quantization", Signal Processing: Image Communication, Vol.8, PP.421-432, 1996.
6. J. Shanbehzadeh and P.O. Ogunbona, "Index Compressed Vector Quantization Based on Index Mapping", IEEE Proc-Visual Image Signal Processing, Vol.144, No.1, PP.31-38, Feb.1997
7. J. Shanbehzadeh, P.O. Ogunbona, and A. Sarafzadeh, "Image Compression Based on Genealogical Relation of TSVQ Indices", Proc. of SPIE: Multimedia Storage and Archiving Systems III, Vol.3527, PP.476-483, Boston, April 1998.
8. J. Shanbehzadeh, A.M. Eftekhari Moghadam, A. Sarafzadeh, F. Mahmoudi and N. Mashtan, "Image Indexing Based on the Genealogical Relation of TSVQ Indices", Proc. Of SPIE: Internet Imaging, Vol.3964, PP.155-159, Jan 2000
9. Yun Q. Shi and Huifang Sun, Image and Video Compression For Multimedia Engineering, New York: CRC press, 2000.
10. N.S. Jayant, P. Noll, "Digital Coding of waveforms Principles and Applications to Speech and Video", Prentice-Hall, Englewood Cliffs, New Jersey, PP.465-485, 1984.