کاربرد روش TLVQ [1] و تبدیل KLT [2] در فشرده سازی درون فریمی [3] تصاویر با نرخ [4] BPP ۰/۰۰۸

ابوالقاسم صیادیان
استادیار

دانشکده برق، دانشگاه صنعتی امیرکبیر

چکیده

روش های چندی کردن برداری VQ، یکی از روش های موفق در فشرده سازی سیگنال دیجیتالی تصاویر می باشند. قابلیت فشرده سازی این روش ها با افزایش طول بلوک های ورودی (از ۴ × ۴ به سمت ۱۶ × ۱۶ و حتی ۳۲ × ۳۲ پیکسل) افزایش چشمگیری می یابد. بالا رفتن بُعد[5] بردارها موجب بروز چهار مشکل عملی می گردد: الف) نیاز به حجم فوق العاده بالای نمونه های آموزشی ب) نیاز به زمان فوق العاده زیاد برای طراحی کتاب کد ج) نیاز به حجم حافظه فوق العاده بالا برای نگهداری کتاب کد د) نیاز به حجم محاسبات فوق العاده بالا در فاز کدینگ.

مشکل (الف) باتوجه به جذابیت و کارآئی این روش در فشرده سازی تصویر قابل رفع است (چون جمع آوری پایگاه داده فقط یك بار انجام می پذیرد). مشکل (د) با طراحی و ایجاد ساختارهای مناسب بر روی کتاب کد و استفاده از روش جستجوی سریع در حد قابل قبولی مرتفع می گردد (مانند ساختارها TSVQ[6] وLSVQ[7].) راه حل مناسبی برای رفع دو مشکل اساسی (ب) و (ج) تاکنون ارائه نشده به قسمی که کیفیت روش ارائه شده قابل مقایسه با روش های UVQ[8] باشد. استفاده همزمان از روش TLVQ و تبدیل KVT راه حلی است که طی این تحقیق برای رفع دو مشکل متذکر ارائه گردیده است. شبیه سازی های انجام پذیرفته نشان می دهد که در نرخ Bpp ۰/۰۰۸ کیفیت PSNR[9] تصویر بازسازی شده حدود dB ۲۲ می باشد.

0.008 BPP Interaframe Image Compression with Two Layer Vector Quantization and KLT Transform

A. Sayadiyan
Assistant Professor

Department of Electrical Engineering Amirkabir
University of Technology

Abstract

Vector quantization is one powerful method for digital image compression. Increase of image subblock size will be reduce the bit per pixel. Two limiting problem for increasing the subblock size are, increase of codebook storage size and needed time for codebook designing. We propose a new method that uses two layer VQ and KLT transform. The result of simulations on many images show the PSNR of new method is 22 dB at 0.008 BPP.

مقدمه

طبق نظریه نرخ ـ اعوجاج[۱] ، درصد فشرده سازی تصاویر (به عنوان یک منبع اطلاعاتی با همبستگی زیاد) با افزایش طول بلوک های زیر تصویر در چندی کننده های برداری افزایش می یابد (تحت شرایط اعوجاج معین). بنابر نظریه فوق، تلاش فراوانی تاکنون به منظور فشرده سازی هر چه بیشتر تصاویر انجام پذیرفته است [۱ الی ۱۰]. استفاده از چندی کننده های برداری با طول بلوک های معمولی (۴×۴ با حداکثر ۸×۸ پیکسل) روشی رایج است. در اغلب کاربردها کتاب کد برای طول بلوک ۴×۴ به صورت ۱۰ بیتی و برای طول ۸×۸ به صورت ۱۲ بیتی طراحی شده مورد استفاده قرار می گیرند [۱ و ۶]. برای بلوک های بزرگتر از ۸×۸، ابتدا با استفاده از تبدیل های مناسب همانند KLT، DCT[۱۱]، DST[۱۲]، WHT[۱۳] و ...، همبستگی های خطی بین پیکسل های داخل هر بلوک را حتی الامکان حذف نموده و در نتیجه اندازه بردار حاصل را حتی المقدور کاهش می دهند (با قبول مقدار اندکی اعوجاج). فرض می کنیم $X=(x_1, ..., x_L)$ بردار ورودی تبدیلDCT و $Y=(y_1, ..., y_L)$ بردار خروجی ضرایب تبدیل باشند. همچنین فرض می کنیم $\delta_Y = (\delta_{y1}, ..., \delta_{yL})$ بردار واریانس ضرایب تبدیل DCT بر روی پایگاه داده های آموزشی باشند. ضرایب تبدیل را بر حسب ترتیب نزولی واریانس آنها مرتب نموده و به صورت $S\delta_Y = (S\delta_{y1}, ..., S\delta_{yL})$ نمایش می دهیم. حال خطای PVMSE[۱۴] را به صورت زیر تعریف می نماییم:

$$pv(p) = \left[\frac{\sum_{i=p+1}^{L} S\delta_{yi}}{\sum_{i=1}^{L} S\delta_{yi}} \right] \times 100 \qquad (1)$$

با افزایش P(بعد بردار کاهش یافته)، خطای PVMSE کاهش می یابد. به هر حال چون هدف اصلی از این تبدیل، کاهش بعد بردار تبدیل یافته (به خاطر اهداف فشرده سازی تصویر) است، ضرورت دارد که P حتی الامکان کوچک در نظر گرفته شود. کوچک کردن غیرمعقول p نیز موجب افزایش غیرقابل قبول خطای PVMSE خواهد شد. بنابراین می بایستی p را طوری انتخاب نماییم که کیفیت تصویر بازسازی شده در حد قابل قبول باقی بماند. بدین لحاظ p را طوری تعیین می نمایند که خطای مزبور در رنج $5 \le pv \le 1$ درصد باقی بماند . انتخاب $L/4 \le p \le L/2$ رنج مناسبی در کاربردهای عملی است. در طی این تحقیق برای L = ۲۵۶ (بلوک های ۱۶×۱۶ پیکسل) p = ۱۲۸ و برای L = ۳۸۶ (بلوک های ۲۴×۱۶ پیکسل) p = ۱۶۰ و برای L =۱۰۲۴ (بلوک های ۳۲ × ۳۲) پیکسل) p = ۲۵۶ به عنوان تعداد ضرایب تبدیل DCT انتخاب گردیدند. حال فرض می کنیم که بردار ضرایب تبدیل یافته با بعد مشخص p به عنوان ورودی سیستم چندی کننده برداری LSVQ[۱۵] باشد. باتوجه به افزایش بعد بردارها در این نوع چندی کننده های برداری، ضروری است که تعداد الگوهای مرجع کتاب کد به اندازه کافی بزرگ انتخاب گردد تا به کیفیت مورد نظر دست یابیم. طبق نتایج شبیه سازی به عمل آمده در طی این تحقیق مناسب است که برای طول بلوک های ۱۶ × ۱۶ کتاب که حداقل ۱۶ بیتی در نظر گرفته شود (در کاربردهای تلفن تصویری). فرض می کنیم هر پارامتر تبدیل DCT، به صورت یک عدد ممیز شناور ذخیره گردند. اگر از چندی کننده های برداری غیرساختار یافته UVQ[۱۶] برای این منظور استفاده نماییم (که از نظر کیفیت بهینه هستند)، جدول (۱) حجم حافظه مورد نیاز برای ذخیره و نگهداری کتاب کد را نشان می دهد. نکته قابل توجه آن است که در کلیه سیستم های عملی LSVQ، حتماً نوعی ساختار مناسب (برای جستجوی سریع) بر روی کتاب کد اعمال می نمایند [۱]. اعمال ساختار مناسب، مستلزم افزایش حجم حافظه مورد نیاز سیستم می باشد. در اغلب روش های VQ ساختار یافته رایج (مانند ساختار درختی TSVQ) حجم حافظه مورد نیاز تا حدود دو برابر کتاب کد اصلی افزایش می یابد [۱ و ۲].

جدول (۱) حجم حافظه مورد نیاز چندی کننده های LSVQ برای کتاب کد ۱۶ بیتی (برحسب مگابایت)

در حالت FSSVQ [۳]		در حالت UVQ	
p =۱۶۰	p =۱۲۸	p =۱۶۰	p =۱۲۸
۸۳/۹	۶۷/۱۲	۴۱/۹۵	۳۳/۵۶

با ملاحظه رقم های جدول (۱)، دلیل اصلی عدم مقبولیت چندی کننده های LSVQ مشخص می گردد. حجم حافظه فوق العاده بالای مورد نیاز چندی کننده های LSVQ، دلیل اصلی عدم مقبولیت و عدم رایج شدن آنها می باشد.

۲ـ چندی کننده های برداری دولایـــه ای

(17) TLVQ

چندی کننده های دولایه ای با اهداف مختلف قبلاً نیز مورد استفاده قرار گرفته اند [۸]. هدف اصلی در استفاده از چندی کننده های TLVQ، یکی کاهش زمان لازم برای طراحی الگوهای مرجع و دیگر کاهش نسبی بار محاسباتی در فاز کدینگ می باشد. فرض می کنیم که N (تعداد الگوهای مرجع کتاب کد) به صورت حاصلضرب دو عدد صحیح $N = N_1 . N_2$ قابل تجزیه باشد. در این صورت از چندی کننده TLVQ به صورت دو ساختار متمایز زیر می توان استفاده نمود [۲ و ۳]: الف) ساختار ضربی PVQ[۲۰] ب) ساختار درختی TVQ[۱۹].

الف) در **ساختار ضربی** ابتدا یک کتاب کد N_1 تایی به عنوان کتاب کد اصلی طراحی می گردد. کلیه بردارهای آموزشی را به یکی از N_1 الگوی مرجع اصلی نسبت می دهند. آنگاه اختلاف بین کلیه بردارهای آموزشی و الگوهای مرجع اصلی متناظر را به دست می آورند. از روی بردارهای اختلاف جدید (که دارای واریانس متوسط کمتری است)، مجدداً یک کتاب کد فرعی با تعداد N_2 الگوی مرجع طراحی می نمایند. این روش از نظر کاهش حجم حافظه مورد نیاز سیستم های LSVQ، بسیار ایده آل می باشد. متأسفانه کیفیت نهایی سیستم از مقدار بهینه (کتاب کد با N عضو) بسیار دور است. برای مثال اگر $N_1 = N_2$ و $N = 2^{16}$ (کتاب کد ۱۶ بیتی) باشد، در این صورت کتاب کد اصلی و فرعی هر دو ۸ بیتی خواهندبود. حجم حافظه مورد نیاز برای ساختار PVQ در این حالت برابر ۰/۲۷ مگابایت می گردد. با مقایسه عدد فوق و اعداد داخل جدول (۱) ملاحظه می گردد که حجم حافظه مورد نیاز در این حالت بسیار ناچیز و در حد بسیار مطلوب قرار دارد. مزیت دوم روش PVQ کاهش بار محاسباتی فاز کدینگ در مقایسه با روش UVQ است. برای مثال تعداد دفعات محاسبه تابع فاصله در فاز کدینگ برای روش UVQ برابر N و در روش PVQ برابر $N_1 + N_2$ بازاء هر بردار تست ورودی است. در نتیجه بار محاسباتی روش PVQ در حدود $(N_1 + N_2) / N \times 100$ درصد روش UVQ خواهد بود. در مثال بالا (با فرض $N_1 = N_2$) بار محاسبات فاز کدینگ روش PVQ در حدود ۰/۸ درصد روش UVQ می گردد. ملاحظه می شود که این مقدار کاهش دربار محاسباتی بسیار چشمگیر است. مزیت سوم روش PVQ که در مقالات و منابع تحقیقاتی کمتر به آن اشاره می شود، کاهش بسیار مطلوب در زمان طراحی کتاب کد است. اگر بردارهای آموزشی زیاد باشد (که در چندی کننده های LSVQ چنین است)، تعداد مراحل بهینه سازی و همچنین تعداد الگوهای مرجع کتاب کد بالا است. بدین لحاظ حجم محاسبات در مرحله طراحی فوق العاده بالا می رود. برای مثال با تعداد بردارهای آموزشی حدود ۱۰۰ برابر تعداد الگوهای مرجع (که برای یادگیری آماری[۲۱] ضروری است)، وقت مورد نیاز (برای کتاب کد ۱۶ بیتی) فقط برای بهینه سازی یک مرحله از الگوریتم LBG[۲۲] [۱] حدود دو ماه بوده است. در صورتی که برای روش PVQ کل فرآیند طراحی الگوهای مرجع در طی کسری از ساعت انجام می پذیرد. ضعف روش PVQ ؛ ضعف عمده روش PVQ پایین بودن کیفیت آن در مقابل روش UVQ است. بنابراین با اینکه روش PVQ از نظر پیاده سازی دارای سه مزیت بسیار جالب است، متأسفانه به صورت فوق الذکر کمتر مورد استفاده قرار می گیرد. در کاربردهای عملی تعداد الگوهای مرجع کتاب کد فرعی به حدود ۴ تا ۶۴ برابر (N_2) افزایش داده می شود. این مطلب به خاطر آن است که به کیفیتی قابل مقایسه با روش UVQ دست یابند [۱ و ۶]. مطلب فوق به این معنی است که برای وصول به یک مقدار از اعوجاج تقریباً یکسان، روش PVQ به نرخی در حدود ۲ تا ۶ بیت بیشتر از روش UVQ نیاز دارد.

ب) ساختار درختی TVQ، همانند روش PVQ ابتدا یک کتاب کد N_1 تایی به عنوان کتاب کد اصلی طراحی می گردد، کلیه بردارهای آموزشی به یکی از N_1 الگوی مرجع اصلی نسبت داده می شوند. آنگاه برای هر الگوی مرجع اصلی مجدداً (با استفاده از بردارهای آموزشی متناظر) یک کتاب کد فرعی با تعداد الگوهای مرجع برابر N_2 طراحی می نماییم. در حقیقت برخلاف روش PVQ که دارای یک کتاب کد فرعی است، در این حالت N_1 کتاب کد فرعی خواهیم داشت که هر کدام دارای N_2 الگوی مرجع هستند. کیفیت ساختار درختی دو لایه ای تقریباً در حد چندی کننده UVQ است [۳] (با اختلاف ۰/۵ dB تا ۰/۱) و بنابراین برای اهداف پیاده سازی بسیار مناسب می باشند. ساختار درختی دو لایه ای علاوه بر تأمین کیفیت مناسب (ضعف اساسی روش PVQ)، دو مزیت عمده روش PVQ یعنی کاهش مناسب بار محاسباتی در مرحله طراحی کتاب کد و فاز کدینگ را دارا می باشد.

متأسفانه حجم حافظه مورد نیاز این ساختار در حد روش چندی کننده UVQ است (حتی اندکی هم بیشتر). بنابراین ساختار درختی به حجم فوق العاده بالای حافظه در مقابل ساختار ضربی PVQ نیازمند است.

مقایسه ساختار PVQ و TVQ: با توجه به مطالب بیان شده و تحت شرایط نرخ مساوی (برای یک کتاب کد ۱۶ بیتی)مزایا و معایب دو روش PVQ و TVQ در مقابل روش UVQ را می توان در جدول (۲) ملاحظه نمود. معیارهای مقایسه عبارتند از: الف) کیفیت ب) بار محاسباتی مورد نیاز در مرحله طراحی کتاب کد ج) بار محاسباتی مورد نیاز برای جستجوی کتاب کد در مرحله کدینگ د) حجم حافظه مورد نیاز برای نگهداری کتاب های کد.

همانطوری که از جدول (۲) مشاهده می شود، هر دو روش PVQو TVQ دارای دو مزیت مشترک (ب و ج) نسبت به روش UVQ می باشند. در دو معیار دیگر هر کدام دارای یک نقطه قوت و یک نقطه ضعف هستند. هدف ما استفاده تلفیقی از دو روش PVQ و TVQ است به قسمی که از نقاط قوت هر کدام استفاده مناسب نموده و نقطه ضعف آنها را توسط روش مناسبی جبران نماییم.

۳ـ۱ـ استفاده از تبدیل KLT برای کاهش حجم حافظه

از میان کلیه تبدیل های خطی ارتونرمال [۲۳]، تبدیل KLT تبدیلی بهینه است. تبدیل هایی مانند DCT و DST و WHT و DFT و ... تبدیل هایی قطعی [۲۴] بوده و نمی توانند از توزیع آماری نمونه های آموزشی برای حذف همبستگی های خطی به نحو شایسته استفاده نمایند. تبدیل KLT بر خلاف تبدیل های قطعی یک تبدیل تصادفی [۲۵] است که از خواص آماری (خطی) بردارهای آموزشی به نحو شایسته ای در کاهش بعد فضای پارامترها استفاده می نماید. فرض می کنیم $\upsilon_1, ..., \upsilon_p$ بردارهای ویژه و $\lambda_1, ..., \lambda_p$ (با فرض $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$) مقادیر ویژه ماتریس کوواریانس بردارهای آموزشی ضرایب تبدیل DCT باشند. با استفاده از رابطه (۲) بعد کاهش یافته p_1 را طوری در نظر می گیریم که خطای نسبی مقادیر ویژه کمتر از ۱ تا ۵ درصد گردد:

$$re(p_1) = \left[\frac{\sum_{i=p_1+1}^{P} \lambda_i^2}{\sum_{i=1}^{P} \lambda_i^2} \right]^{1/2} \times 100 \qquad (۲)$$

در شبیه سازی های انجام پذیرفته طی این تحقیق، مشخص شده است که با انتخاب $p_1 = ۸۰$ به خطای کمتر از ۲/۸۱ درصد می رسیم که مقدار قابل قبولی است. ملاحظه می گردد که اگر از تبدیل KLT در لایه اول استفاده نماییم، به کاهشی در حدود ۳۷ درصد حجم حافظه مورد نیاز سیستم خواهیم رسید. برای دسترسی به این مقدار کاهش در حجم حافظه، مجبور به صرف بار محاسباتی اضافی $2p_1.p$ (عمل ضرب و جمع) بازاء هر بردار ورودی (به خاطر انجام تبدیل KLT) می باشیم. باتوجه به نتایج فوق به نظر می رسد که اعمال تبدیل KLT در لایه اول به نتایج خیلی خوب در کاهش حجم حافظه منجر نمی گردد.

جدول (۲) مقایسه مزایا و معایب سه روش PVQ و TVQ و UVQ

نوع مقایسه / نوع چندی کننده	با معیار(الف)	با معیار(ب)	با معیار(ج)	با معیار(د)
UVQ	بهینه	خیلی بالا	خیلی بالا	خیلی بالا
PVQ	ضعیف	خیلی پایین	پایین	خیلی کم
TVQ	نزدیک به بهینه	پایین	پایین	خیلی بالا

۳ـ روش اعمال تبدیل KLT

در طی این تحقیق متوجه شدیم که اگر تبدیل KLT در لایه دوم بر روی بردارهای مرجع ضرایب تبدیل DCT اعمال گردد، به نتایج نسبتاً جالبی در فشرده سازی پارامترهای کتاب کد دست می یابیم. برای این منظور ابتدا N_1 ماتریس کوواریانس متناظر با بردارهای آموزشی ضرایب تبدیل DCT, N_1 کتاب کد فرعی را تعیین می نماییم. آنگاه از روی این ماتریس ها، N_1 دسته از مقادیر ویژه و بردارهای ویژه را محاسبه می نماییم. بر مبنای رابطه (۲)، بعد بردارهای کاهش یافته p_{i1} $(j = 1\ N_1)$ را طوری تعیین می کنیم که خطای نسبی حاصل بر روی بردارهای ضرایب تبدیل DCT کمتر از ۱ تا ۵ درصد گردد. شبیه سازی های انجام پذیرفته طی این تحقیق نشان می دهد که انتخاب $۲۰ \leq p_{1j} \leq ۱۶$ برای $p = ۱۲۸$ (تعداد ضرایب تبدیل DCT بلوک ۱۶ × ۱۶ پیکسل) بعد کاهش یافته مناسبی می باشد. باتوجه به ملاحظات عملی، بعد یکسان برای

کلیه بردارهای کاهش یافته ضرایب تبدیل DCT انتخاب شده و برابر $p_{1j} = ۱۶$ در نظر گرفتیم (به خطای نسبی متوسط ۲/۸۵ درصد منجر می‌شود). در این حالت به $۱۶N_1$ بردار p پارامتری اضافی (نسبت به روش UVQ) نیاز خواهیم داشت (برای نگهداری ۱۶ بردار ویژه N_1 کتاب کد فرعی). با این وجود حجم حافظه مورد نیاز در این حال از ۳۳/۵۶ مگابایت به ۶/۴۳ مگابایت کاهش می‌یابد (چنانچه از ساختار درختی دو لایه‌ای استفاده نماییم). یعنی استفاده از تبدیل KLT بر روی ضرایب تبدیل DCT در لایه دوم موجب کاهش حجم حافظه مورد نیاز به مقدار ۸۱٪ شده است. با اینکه حجم حافظه مورد نیاز برای کتاب کد ۱۶ بیتی از ۳۳/۵۶ مگابایت به ۶/۴۳ مگابایت کاهش یافته (به کمتر از ۵/۱ تقلیل یافته)، ولی متأسفانه هنوز برای کاربردهای عملی مناسب به نظر نمی‌رسد. برای کاهش بیشتر حجم حافظه از ساختار PVQ با اصلاحاتی به شرح زیر استفاده نمودیم. ابتدا یک کتاب کد (۱۲ـ۱۰) بیتی فرعی سرتاسری [۲۶] با استفاده از کلیه بردارهای با بعد کاهش یافته ($p_1 = ۱۶$) (مربوط به کلیه کتاب های کد فرعی) طراحی مینماییم (به جای ۲۵۶ کتاب کد فرعی ۸ بیتی). آنگاه برای هر الگوی مرجع در کتاب های کد فرعی یک ایندکس متناظر در کتاب کد فرعی سرتاسری تعیین می نماییم. الگوی متناظر تعیین شده در کتاب کد فرعی سرتاسری می بایستی دارای کمترین فاصله با الگوی مورد آزمایش باشد. اگر هر ایندکس توسط حداکثر ۲ بایت نمایش داده شود، برای ذخیره کردن $N = ۲^{۱۶}$ الگوی مرجع فرعی در این حالت فقط به ذخیره کردن 2N بایت یعنی ۰/۱۲۸ مگابایت نیاز خواهیم داشت. ملاحظه می‌شود که عدد فوق برای یک کتاب کد ۱۶ بیتی بسیار جذاب و جالب توجه می‌باشد. البته در این حالت برای ذخیره کردن کتاب کد فرعی سرتاسری و بردارهای ویژه به حافظه اضافی نیاز داریم. باتوجه به بحث فوق، حجم کل حافظه مورد نیاز در این حالت ۲/۶۳ مگابایت می‌گردد. ملاحظه می‌شود که حجم حافظه مورد نیاز نسبت به روش UVQ حدود ۹۲/۲ درصد کاهش یافته است. از این مقدار حدود ۲/۲۳ مگابایت (یعنی ۸۱/۸ درصد) فقط جهت ذخیره نمودن کتاب کد اصلی و N1 مجموعه از بردارهای ویژه می‌گردد.

٤ ـ نتایج شبیه سازی

هدف اصلی این تحقیق بررسی کارآیی روش پیشنهادی برای فشرده‌سازی درون فریمی تصویر در کاربردهای تلفن تصویری بوده است. برای مقایسه عملکرد روش‌ها از مقدار PSNR[۲۸] بر حسب dB استفاده نمودیم. اگرفرض کنیم $X(n)=[x_1(n), \dots, x_p(n)]$ یک نمونه از بردارهای غیرچندی شده و $Y(n) = [y_1(n), \dots, y_p(n)]$ یک نمونه از بردارهای چندی شده متناظر با $X(n)$ باشد، در این صورت مقدار PSNR را بر حسب dB از رابطه زیر تعیین نمودیم (L تعداد کل بردارهای تست برای ارزیابی است):

$$PSNR = \frac{1}{L}\sum_{n=1}^{L} 10 \log\left[\frac{\sum_{i=1}^{P} x_i^2(n)}{\sum_{i=1}^{P}(x_i(n) - y_i(n))^2}\right] \quad (۳)$$

آزمون‌ها برای سه حالت زیر انجام پذیرفته است: الف) کتاب کد دو لایه‌ای با ساختار درختی TVQبدون استفاده از تبدیل KLT ب) کتاب کد دو لایه‌ای با ساختار درختی TVQ و با استفاده از تبدیل KLT و ۲۵۶ کتاب کد فرعی ج) کتاب کد دو لایه‌ای با ساختار درختی TVQ با استفاده از تبدیل KLT و یک کتاب کد فرعی سرتاسری ۱۲ بیتی.

جدول (۳) نتایج شبیه سازی حاصل را برای دو نوع آزمون نشان می‌دهد. در آزمون اول نمونه های تست از میان مجموعه نمونه های آموزشی انتخاب گردیدند. در آزمون دوم نمونه‌ای تست خارج از مجموعه نمونه های آموزشی انتخاب شدند. در هر دو آزمون تعداد بلوک های تست ۳۰۰۰۰ بردار بوده است. (۱۰۰ تصویر ۲۴۰ × ۳۲۰ با بلوک های ۱۶ × ۱۶ پیکسل).

با بررسی جدول (۳) ملاحظه می‌شود که با قبول حداکثر ۲dB، می‌توان یک سیستم عملی (با حجم حافظه قابل قبول) برای فشرده‌سازی درون فریمی با نرخ فشرده‌سازی بالا یعنی ۰/۰۰۸ bpp با استفاده از روش پیشنهادی طراحی نمود. برای مشاهده نتایج کیفی، دو تصویر از آزمون اول و دوم در شکل (۱) و (۲) نشان داده شده‌اند. شکل (۱)، یک نمونه از تصویربازسازی شده است وقتی که تصویر اصلی در مجموعه نمونه های آموزشی حضور داشته است. شکل (۲) یک نمونه از تصویر بازسازی شده است، وقتی که تصویر اصلی در مجموعه نمونه های آموزشی حضور نداشته است.

با ملاحظه نتایج کمّی و کیفی حاصل بر روی تعداد زیادی از تصاویر، به نظر می‌رسد که روش پیشنهادی دارای پتانسیل مناسبی برای فشرده‌سازی درون فریمی تصاویر با نرخ خیلی پایین در کاربردهای تلفن تصویری می‌باشد.

جدول (۳) نتایج شبیه سازی برای روش های مختلف چندی کننده برداری دو لایه ای

PSNR (dB) / حالات تست	آزمون اول	آزمون دوم
حالت الف	۲۵/۲۸	۲۴/۱۷
حالت ب	۲۴/۳۱	۲۳/۰۵
حالت ج	۲۳/۴۷	۲۲/۱۲

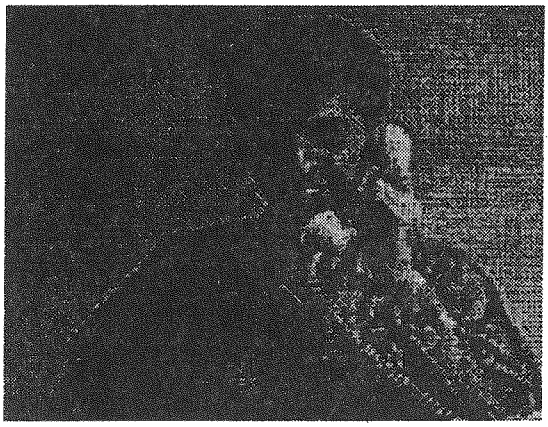
۵ـ جمع بندی و نتیجه گیری

در این تحقیق نشان داده ایم که با استفاده از نظریه چندی کننده های برداری دو لایه ای می توان به فشرده سازی بدرون فریمی با راندمان خیلی بالا دست یافت (با نرخ ۰/۰۰۸ bpp و مقدار PSNR متوسط dB ۲۳/۴۷). برای این منظور از تلفیق دو نظریه چندی کردن برداری ضربی و درختی به نحو مناسبی استفاده نمودیم. همچنین برای کاهش حجم حافظه مورد نیاز در قسمت کتاب های کد فرعی از تبدیل KLT در لایه دوم استفاده نمودیم که بر روی بردارهای ضرایب تبدیل DCT آموزشی اعمال گردیده است. نتایج شبیه سازی با دیتاهای زیاد در کاربردهای تلفن تصویری نشان می دهد که کارآئی روش پیشنهادی هم از نظر کمّی و هم از نظر کیفی و هم از نظر پیاده سازی قابل توجه می باشد.

زیرنویس ها

1 - Two Layer Vector Quantization
2 - Karhunen - Loeve Transform
3 - Intera frame
4 - Bit Per Pixel
5 - Dimension
6 - Tree Structure VQ
7 - Lattice Structure VQ
8 - Unstructural VQ
9 - Proportional Signal to Noise Ratio
10 - Rate Distortion Theory
11 - Discrete Cosine Transform
12 - Discrete Sine Transform
13 - Walsh Hadamard Transform
14 - Proportional Variance Mean Square Error
15 - Large Size VQ
16 - Fast Search Structure VQ
17 - Two Layer VQ
18 - Unstructural VQ
19 - Product VQ
20 - Statistical Learning
21 - Tree VQ
22 - Linde, Buzo, Gray
23 - Orthonormal
24 - Deterministic
25 - Stochastic
26 - Global
27 - Proportional Signal to Noise Ratio

شکل (۱) الف ـ تصویر اصلی



ب) تصویر بازسازی شده

شکل (۲) الف) تصویر اصلی

ب) تصویر بازسازی شده

مراجع

[1] V. S. Sitaram, C. M. Huang P. D. Israelsen, "Efficient Codebooks for Vector Quantization Image Compression with an Adaptive Tree Search Algorithm", IEEE Trans on Commu., Vol 42, No. 11. Novem, 1994.

[2] Y. C. Lin, S.C. Tal, "A Fast Vector Quantization Encoder for Video Coding", IEEE Trans on Consumer Electronics, Vol. 40, No. 3, August 1992.

[3] N. Moayeri, D. L., Nouhoff, "Time - Memory Tradeoffs in Vector Quantizer Codebook Searching Based on Decision Trees", IEEE Trans. on Speech and Audio Processing Vol. 2, No. 4, Oct. 1994.

[4] K. L. Oehler, R. M. Gray, "Combining Image Compression and Classification using Vector Quantization", IEEE Trans on Pattern Analysis and Machine Intelligence, Vol. 17, No. 5, May 1995.

[5] C. H. Lee, L. H. Chen, "A Fast Search Algorithm for Vector Quantization using Mean Pyramids of Codewords", IEEE Trans. on Commu. Vol. 43, No. 2/3/4, Feb./Mar./Apri. 1995.

[6] M. A. Ghafourian, Ch. M. Huans, "Comparison Between Several Adaptive Search Vector Quantization Schemes and JPEG Standard for Image compression" IEEE Trans. on Commu., Vol. 43, No. 2/3/4, Feb./Mar./April 1995.

[7] K. T. LO, J. Feng, "Predictive Mean Search Algorithms for Fast VQ Encoding of Image", IEEE Trans. on Commu, Vol. 43, No. 12, Decem. 1995.

[8] S. C. Toi, C. C. Loi, Y. C. Lin, "Two Fast Nearest Neighbor Searching Algorithm for Image Vector Quantization", IEEE Trans. on Commu Vol. 40, No. 12. Decem 1996.

[9] S. Y. Choi, S. L. O chae, "Fast Vector Quantizer Usig Multiple Sorted Index Tables" IEEE Proc. on ICASSP, 1996.

[10] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantization Design", IEEE Trans. On Commu, , Vol. Com 28, Jan. 1980.