پیاده‌سازی فشرده‌سازی تلفاتی تصویر توسط موجک CDF(۲،۲) در CPLD

عباس‌علی لطفی نیستانک، محمد محقق حضرتی، مهدی شارعی و نرگس احمیدی

چکیده: در این مقاله پس از مقایسه روشهای فشرده‌سازی تصویر از قبیل روش BTC، روش اهرام گاوسی، روش SVD، روش تبدیل موجک و یا بطور خاص CDF(۲،۲) به پیاده‌سازی سخت‌افزاری فشرده‌سازی تصویر به روش موجک CDF(۲،۲) پرداخته شده است. طراحی ارائه‌شده نشان‌دهنده این است که سازمان‌دهی مناسب داده‌ها (روش تقسیم‌بندی) و استفاده از خط لوله و پردازش موازی در بهینه‌سازی سخت‌افزاری مدار تاثیر زیادی دارد. در حقیقت هدف اصلی، ایجاد کارایی و سرعت بیشتر در CPLD ساخت شرکت Xilinx به نام XC۹۵۷۲ می‌باشد. جزئیات طراحی کد گذار و همچنین نتایج بدست آمده نیز در پایان ارائه شده‌اند. نتایج مقایسه روش‌های مختلف فشرده‌سازی تصویر می‌تواند برای یک کاربر الگوی مناسبی جهت استفاده از روش بهینه را با توجه به نوع مسئله ارائه دهد. برای شبیه‌سازی از نرم‌افزار MATLAB و همچنین از C++ استفاده شده است.

کلید واژه: **اهرام گاوسی، تبدیل موجک گسسته، تقسیم‌بندی، فشرده‌سازی تصویر، BTC، CDF(۲،۲)، CPLD، SVD.**

۱- مقدمه

امروزه سیستم‌های پردازش تصویر می‌توانند تصاویر خام را با قدرت تفکیک و کیفیت دلخواه فشرده کنند و در نتیجه به سطوح متفاوتی از فشرده‌سازی برحسب کاربرد دست یابند. این نیاز همواره وجود داشته است که نسبت فشرده‌سازی را بتوان بسته به کاربرد مورد نظر تغییر داد. به عنوان مثال، می‌توان به انتقال بلادرنگ تصویر و ویدئو از طریق شبکه پکت سوئیچ اشاره کرد. تصاویر با کیفیت بالا که از استودیوها، عکس‌های پزشکی، اطلاعات لزره‌شناسی، تصاویر ماهواره‌ای و تصاویر اسکن‌شده از نوشته‌های خطی به منظور حفظ آثار تاریخی بدست می‌آیند، به علت سایز عظیم داده‌ها به منظور ذخیره و نگهداری و همچنین برای ارسال از طریق تجهیزات مخابراتی با پهنای باند محدود، نیاز به فشرده‌سازی بدون تلفات دارند و حتی‌الامکان فشرده‌سازی با تلفات در مورد آنها باید به نحوی صورت گیرد که خطای دیده‌شده توسط چشم قابل صرف نظر باشد [۱]. بدین منظور باید قدرت تفکیک و نرخ بیت تصاویر فشرده‌شده تا حدی که چشم تشخیص ندهد کاهش یابد. جدول ۱ نمونه‌هایی از انواع داده‌های چندرسانه‌ای همراه با فضای ذخیره‌سازی لازم برای آنها را بدون انجام فشرده‌سازی نشان می‌دهد. رشد کنونی کاربردهای داده‌های دیجیتالی صوتی، تصویری و ویدئویی (چندرسانه‌ای) نه تنها نیازمند روشهای کارآمدتر برای کد گذاری سیگنال‌ها و تصاویر می‌باشند، بلکه فشرده‌سازی چنین سیگنال‌هایی برای کاهش میزان ذخیره‌سازی داده‌ها و ارسال مخابراتی آنها نیز ضروری است.

جدول ۱: انواع داده‌های چندرسانه‌ای و فضای مورد نیاز برای ذخیره آنها

Multimedia Data	Size/ Duration	Bits/Pixel (bpp) or Bits/ Sample (bps)	Uncompressed Size
A page of text	۱۱"×۸٫۵"	Varying resolution	۱۶–۳۲ kbits
Telephone quality speech	۱ sec	۸ bps	۶۴ kbits
Grayscale Image	۵۱۲×۵۱۲	۸ bpp	۲٫۱ Mbits
Color Image	۵۱۲×۵۱۲	۲۴ bpp	۶٫۲۹ Mbits
Medical Image	۲۰۴۸×۱۶۸۰	۱۲ bpp	۴۱٫۳ Mbits
SHD Image	۲۰۴۸×۲۰۴۸	۲۴ bpp	۱۰۰ Mbits
Full-motion Video	۶۴۰×۴۸۰، ۱۰ sec	۲۴ bpp	۲٫۲۱ Gbits

تبدیل گسسته موجک یا DWT یک ابزار کارآمد و مفید برای کاربردهای پردازش تصویر و سیگنال می‌باشد [۲].

در سالهای اخیر، استفاده از تبدیل‌های موجک در فشرده‌سازی تصویر بدون تلفات [۳] و همچنین با تلفات در بسیاری از استانداردهای جدید از جمله استاندارد فشرده‌سازی JPEG۲۰۰۰ موفقیت‌آمیز بوده است [۴].

برای مقایسه از پارامترهای مختلفی نظیر میانگین مجذور خطا MSE[1] [۱] و حداکثر سیگنال به نویز PSNR[2] [۱] که از روابط زیر بدست می‌آیند، استفاده شده است

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[\hat{X}(i,j) - X(i,j)] \qquad (1)$$

$$PSNR = 10\log_{10}\left[\frac{256\times256}{MSE}\right] \text{dB}$$

1. Mean Square Error
2. Peak Signal to Noise Ratio

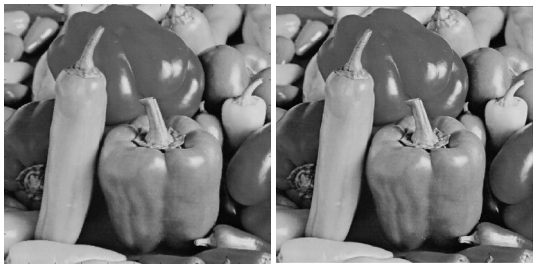
این مقاله در تاریخ ۴ تیر ماه ۱۳۸۴ دریافت و در تاریخ ۲۵ فروردین ماه ۱۳۸۵ بازنگری شد.

عباس‌علی لطفی نیستانک، پژوهشکده برق جهاد دانشگاهی، نارمک، تهران، کدپستی ۱۶۸۴۴، ایران (email: alotfi@iust.ac.ir).

محمد محقق حضرتی، دانشگاه آزاد اسلامی، واحد شهر ری، شهر ری، ایران. (email: mehdi1107@yahoo.com).

مهدی شارعی، دانشگاه آزاد اسلامی، واحد شهر ری، شهر ری، ایران. (email: mo_hzi@yahoo.com).

نرگس احمیدی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران. (email: narges_ahmidi@yahoo.com)



(ب) (الف)

شکل ۱: (الف) تصویر اصلی و (ب) تصویر فشرده‌شده با الگوریتم **BTC**.

جدول ۲: نتایج پیاده‌سازی الگوریتم BTC

زمان اجرای فشرده‌سازی (t) برحسب ثانیه در MATLAB	۱۴٫۲۶۰۰
خطای MSE	۳۴٫۹۹۰
PSNR برحسب dB	۳۲٫۴۶۶۵
نرخ بیت (bits/pixel)	۲

در (۱)، $M = N = ۲۵۶$، X تصویر اصلی و $\hat{X}$ تصویر فشرده‌شده می‌باشند.

در این مقاله پس از مروری کوتاه بر روی روش‌های کلاسیک فشرده‌سازی تصویر از جمله BTC[1]، اهرام گوسی و SVD[2] و موجک یا بطور خاص CDF(۲،۲)[3] [۵] و پیاده‌سازی آنها از لحاظ نرم‌افزاری بر روی یک تصویر نمونه، به ارزیابی و مقایسه این روشها پرداخته شده است. سپس روش موجک به عنوان یک روش کارا و ساده از جهت پیاده‌سازی بر روی یک تراشه CPLD به صورت سخت‌افزاری پیاده شده است. پلت‌فرم مورد استفاده کامپیوتر PentiumIII (۸۰۰ MHz) می‌باشد.

CPLDها یکی از کارآمدترین ابزارهای مدل‌سازی سخت‌افزاری می‌باشند و می‌توان آنها را با زبان سخت‌افزاری VHDL به نحوی برنامه‌ریزی نمود که کاربردهای مورد نظر را بدون هزینه‌هایی که مربوط به ساخت مدارهای مجتمع سفارشی است برآورده سازند. امروزه روشهای مختلف فشرده‌سازی بر روی تراشه‌های قابل برنامه‌ریزی پیاده شد و قابلیت بالای آنها بخوبی اثبات شده است. در مرجع [۶] یک نمونه از این پیاده‌سازی ارائه شده است.

۲- فشرده‌سازی تصویر با روش BTC

الگوریتم فشرده‌سازی تصویر BTC شامل مراحل زیر است:

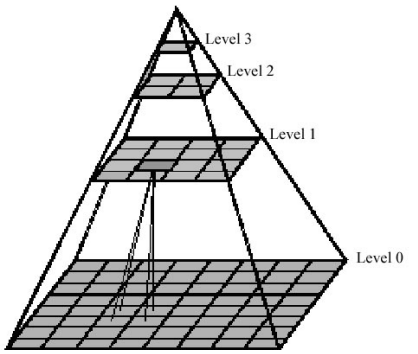
الف) ابتدا تصویر به بلوک‌های غیر همپوشان تقسیم می‌شود. سایز هر بلوک می‌تواند ۴×۴ یا ۸×۸ (برحسب پیکسل) باشد. ابتدا متوسط سطح خاکستری بلوک ۴×۴ محاسبه می‌شود [۷]

$$m = \frac{۱}{۱۶}\sum\sum x(i,j) \qquad (۲)$$

که $x(i,j)$ نمایانگر پیکسل‌ها در یک بلوک می‌باشد.

ب) سپس پیکسل‌ها برحسب اینکه سطح خاکستری آنها بیشتر یا کمتر از متوسط فوق باشد به دو دسته طبقه‌بندی می‌شوند که متوسط سطح خاکستری آنها عبارتست از [۷]

1. Block Truncation Coding
2. Singular Value Decomposition
3. Cohen-Debuchies-Feaveu (2,2)



شکل ۲: تصاویر فیلترشده بر روی یکدیگر قرار گرفته به شکل یک هرم.

$$m_u = \frac{۱}{K}\sum_{x(i,j)<m} x(i,j)$$
$$m_l = \frac{۱}{۱۶-K}\sum_{x(i,j)\geq m} x(i,j) \qquad (۳)$$

که K تعداد پیکسل‌هایی است که سطح خاکستری آنها بیشتر از m باشد.

ج) همچنین یک بلوک باینری (b) برای طبقه‌بندی پیکسل‌ها نیاز است. پیکسلی که سطح خاکستری‌اش بیشتر از m است با ۱ و پیکسلی که سطح خاکستری‌اش کمتر یا مساوی m است با صفر نشان داده می‌شود.

د) قسمت کد گذار m_l، m_u و b را در یک فایل ذخیره می‌کند. فرضاً اگر m_l و m_u هشت‌بیتی باشند، مجموع تعداد بیت‌های مورد نیاز برای نشان دادن یک بلوک، $۳۲ = ۱۶+۸+۸$ خواهد بود. بنابراین نرخ بیت برای الگوریتم BTC در این حالت bits/pixel ۲ خواهد بود. بدین مفهوم که میزان فشرده‌سازی تصویر برابر ۴ به ۱ خواهد شد (نرخ بیت تصویر اصلی برابر bits/pixel ۸ می‌باشد).

ه) در قسمت کد گشایی، بلوک‌ها با جایگزینی یکها بجای m_u و صفرها بجای m_l بازسازی می‌شوند.

نتایج شبیه‌سازی الگوریتم BTC با نرم‌افزار MATLAB و بر روی یک تصویر ۲۵۶×۲۵۶ در شکلهای ۱ و جدول ۲ نشان داده شده است. در این حالت نسبت فشرده‌سازی ۴:۱ است.

۳- فشرده‌سازی تصویر بر اساس اهرام گاوسی

نحوه محاسبه هرم گاوسی[4] بدین صورت است که تصویر اصلی با هسته گاوسی[5] امتزاج می‌شود. تصویر بدست آمده یک نسخه فیلترشده پایین‌گذر تصویر اصلی می‌باشد. فرکانس قطع را می‌توان با پارامتر δ کنترل کرد. سپس لاپلاسین که همان تفاوت میان تصویر اصلی و تصویر فیلترشده است، محاسبه می‌شود. این روند ادامه می‌یابد تا مجموعه‌ای از تصاویر فیلترشده میان‌گذر بدست آید (چون هر تصویر تفاوت بین دو سطح هرم گاوسی است). بنابراین همانطور که در شکل ۲ نشان داده شده است هرم لاپلاسین مجموعه‌ای از فیلترهای میان‌گذر است.

فرض کنید که $\mathbf{I}(x,y)$ ماتریس تصویر اصلی مورد نظر است. هرم گاوسی روی تصویر $\mathbf{I}$ به صورت زیر تعریف می‌شود

$$G_۰(x,y) = \mathbf{I}$$
$$G_{i+۱}(x,y) = REDUCE(G_i(x,y)) \qquad (۴)$$

4. Gaussian Pyramids
5. Gaussian Kernel

(ب) (الف)

شکل ۳: (الف) تصویر اصلی و (ب) تصویر فشرده‌شده با الگوریتم اهرام گاوسی.

عملگر REDUCE با امتزاج تصویر با فیلتر پایین‌گذر گاوسی صورت می‌گیرد. این فیلتر به نحوی طراحی شده است که پیکسل مرکزی، وزن بیشتری نسبت به پیکسل‌های مجاور دارد و عبارات باقیمانده به نحوی انتخاب می‌شوند که مجموع آنها ۱ باشد. هسته گاوسی توسط فرمول زیر تعریف می‌شود [۸]

$$w(r,c) = w(r)\,w(c) \qquad (۵)$$

که $w(r)$ عبارتست از

$$\left(\frac{۱}{۴}-\frac{a}{۲} \quad \frac{۱}{۴} \quad a \quad \frac{۱}{۴} \quad \frac{۱}{۴}-\frac{a}{۲}\right)$$

a مقداری بین ۰٫۳ تا ۰٫۶ درنظر گرفته می‌شود. خطای پیش‌بینی $L_0(x,y)$ به صورت زیر است

$$L_i(x,y) = G_i(x,y) - EXPAND(G_{i+1}(x,y)) \qquad (۶)$$

عملگر $EXPAND$ به صورت زیر تعریف می‌شود

$$G_{i+1}(x,y) = ۴\sum_{m=-۲}^{۲}\sum_{n=-۲}^{۲} w(m,n)\,G_i\left(\frac{x-m}{۲},\frac{y-n}{۲}\right) \qquad (۷)$$

تنها عباراتی که به ازای آنها عبارات $(x-m)/۲$ و $(y-n)/۲$ منجر به عدد صحیحی شود در مجموع (sum) درنظر گرفته‌شده و بجای G_0، L_0 و L_1 کد می‌شوند. این مطلب منجر به فشرده‌سازی داده خالص می‌شود چون:

- L_0 بسیار ناهمبسته[1] است پس می‌توان آن را پیکسل به پیکسل و با تعداد بیت بسیار کمتر نسبت به G_0 نمایش داد.
- G_1 فیلتر پایین‌گذر بوده که می‌توان آن را با نرخ نمونه‌برداری کمتری کد کرد. با تکرار این روند می‌توان به فشرده‌سازی بیشتر داده دست یافت.

با تکرار این مراحل می‌توان به‌ترتیب به تصاویر L_0، L_1، L_2، ... و L_n دست یافت. حال اگر تصور کنیم که این تصاویر بر روی یکدیگر قرار گرفته باشند، نتیجه یک ساختار هرم‌شکل خواهد بود. بنابراین از هرم لاپلاس می‌توان برای نمایش یک سری تصاویر فیلترشده میان‌گذر استفاده کرد که هر یک با نرخ نمونه‌برداری کمتری نسبت به تصویر قبلی بدست آمده‌اند. این روش به خاطر سادگی محاسبات و سرعت بیشتر، در پردازش تصویر، شناخت الگوی تصاویر و فشرده‌سازی تصویر مورد استفاده قرار می‌گیرد.

نتایج شبیه‌سازی الگوریتم اهرام گاوسی با نرم‌افزار MATLAB و بر روی یک تصویر ۲۵۶×۲۵۶ در شکل ۳ و جدول ۳ نشان داده شده‌است. در این حالت نسبت فشرده‌سازی ۲:۱ است.

جدول ۳: نتایج پیاده‌سازی روش اهرام گاوسی

زمان اجرای فشرده‌سازی (t) برحسب ثانیه در MATLAB	۱۱٫۰۳۶۰
خطای MSE	۱۰۸٫۰۲۱۴
PSNR برحسب dB	۲۶٫۸۲۳۶
نرخ بیت (bits/pixel)	۴

٤- فشرده‌سازی تصویر بر اساس SVD

اصولاً نمایش یک تصویر توسط یک ماتریس دوبعدی $m\times n$ صورت می‌پذیرد. در این روش ابتدا الگوریتم SVD به این ماتریس اعمال می‌شود تا ماتریس‌های $\mathbf{U}$، $\mathbf{S}$ و $\mathbf{V}$ بدست آیند. $\mathbf{S}$ یک ماتریس قطری $n\times m$ است که عناصر غیر صفر آن روی قطر ماتریس، نشان‌دهنده رتبه ماتریس تصویر اصلی می‌باشند. مفهوم اساسی روش فشرده‌سازی تصویر SVD استفاده از تعداد رتبه‌های کمتری برای تقریب‌زدن ماتریس اصلی (ماتریس تصویر مورد نظر) می‌باشد [۹].

این عملیات را می‌توان برای تصاویر اصلی و بازسازی‌شده به ترتیب توسط (۸) و (۹) نمایش داد

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T \qquad (۸)$$
$$\mathbf{S} = diag\,(r_1, r_2, ..., r_k, ۰, ..., ۰)$$

$$\mathbf{A}_1 = \mathbf{U}\mathbf{S}_1\mathbf{V}^T \qquad (۹)$$
$$\mathbf{S}_1 = diag\,(r_1, r_2, ..., r_k)$$

که در این معادلات $\mathbf{U}$ ماتریس $m\times n$، $\mathbf{V}$ ماتریس $n\times m$، $\mathbf{U}_1$ ماتریس $m\times k_1$ و $\mathbf{V}_1$ ماتریس $k_1\times n$ است. بدین ترتیب

$$\left|\mathbf{A}-\mathbf{A}\right|_2 = r_k + ۱ \qquad (۱۰)$$

تصاویر تست‌شده در این مقاله همگی ماتریس‌هایی مربعی $(m\times m)$ می‌باشند. این تصاویر هریک $۲۵۶\times ۲۵۶ = ۵۲۴۲۸۸$ Bytes حافظه برای ذخیره‌شدن نیاز دارند. با اعمال SVD و $K = ۶۰$ نسبت فشرده‌سازی تقریباً برابر ۴ به ۱ می‌شود و حافظه اشغال‌شده حدود $۲mk + k = ۳۰۷۸۰$ بایت خواهد بود که یک‌چهارم حافظه اشغالی توسط ماتریس تصویر اصلی است. لازم به ذکر است که k باید مقداری کمتر از $m^2/(۱+۲m)$ داشته باشد. که در این تصاویر مقدار k باید عددی کمتر از ۱۲۷ باشد. نتایج شبیه‌سازی الگوریتم SVD بر روی یک تصویر ۲۵۶×۲۵۶ در شکل ۴ و در جدول ۴ نشان داده شده است. نسبت فشرده‌سازی ۴:۱ است.

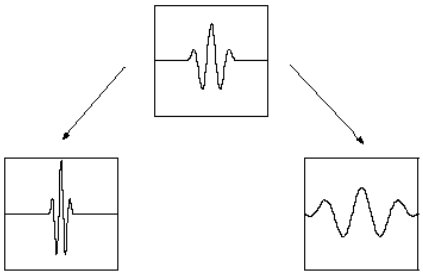
٥- تبدیل موجک گسسته

تا چندی پیش تبدیل فوریه برای تحلیل و بازسازی سیگنال مورد استفاده قرار می‌گرفت. اما تبدیل فوریه هیچ اطلاعات محلی از سیگنال اصلی را شامل نمی‌شود. سپس تبدیل فوریه زمان کوتاه (STFT یا تبدیل گابور) ارائه شد که بطور یکنواختی از صفحه زمان- فرکانس نمونه‌برداری می‌کرد. در نهایت با ارائه تبدیل موجک، تفکیک زمانی خوب و تفکیک فرکانسی ضعیف در فرکانس‌های بالا و تفکیک فرکانسی خوب و تفکیک زمانی ضعیف در فرکانس‌های پایین بدست آمد.

1. Uncorrelated

(ب) (الف)

شکل ۴: (الف) تصویر اصلی و (ب) تصویر فشرده‌شده با الگوریتم SVD.



شکل ۵: موجک مادر و نسخه‌های شیفت‌یافته (مقیاس داده شده) آن.

جدول ۴: نتایج پیاده‌سازی الگوریتم SVD

زمان اجرای فشرده‌سازی (t) برحسب ثانیه در MATLAB	۸۳٫۹۱۱۰
خطای MSE	۵۲٫۹۱۱
PSNR برحسب dB	۳۰٫۶۵۸۱
نرخ بیت (bits/pixel)	۲

در سال ۱۹۸۶، Meyer و Mallat دریافتند که تجزیه و بازسازی موجک ارتونرمال را می‌توان در غالب تحلیل سیگنال با قدرت تفکیک‌های مختلفی پیاده‌سازی کرد. در حقیقت ما قادریم سیگنال را در چندین باند فرکانسی تحلیل کنیم. اصول موجک، مجموعه‌ای از موجهاست که توسط مقیاس‌هایی از موجک مادر مطابق با شکل ۵ بدست می‌آیند.

در حوزه پردازش تصویر، تبدیل موجک در کاربردهایی از قبیل فشرده‌سازی تصویر و بازسازی آن، استخراج قسمت‌های خاص تصویر با کیفیت بالاتر، حذف نویز و ثبت تصویر مورد استفاده قرار می‌گیرد.

فشرده‌کننده‌های با تلفات که بر اساس موجک می‌باشند، از سه مولفه اصلی تشکیل می‌شوند:

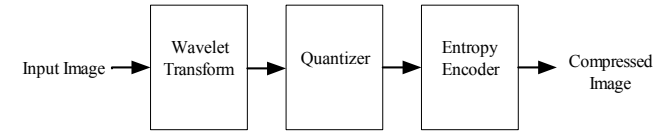
۱- یک فیلتر بانک موجک که تصویر را به ضرایب موجک تبدیل می‌کند.

۲- این ضرایب کوانتیزه می‌شوند.

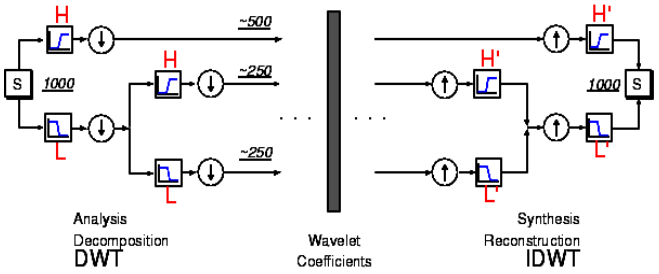
۳- در نهایت فشرده‌ساز آنتروپی روی آنها اعمال می‌شود تا تصویر فشرده‌شده مطابق شکل ۶ حاصل شود.

انتخاب فیلتر و در نتیجه خواص موجک متناظر با آن به الگوریتم فشرده‌سازی (روش کوانتیزاسیون و اختصاص بیتها) بستگی دارد. این مقاله از الگوریتم فشرده‌سازی zerotree که برای اولین بار توسط شاپیرو [۱۰] ارائه شد، بهره می‌گیرد.

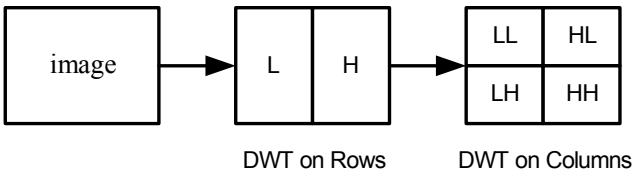
در اینجا ما تنها تبدیل‌های موجک را برای پردازش و آنالیز داده‌های تصویر دوبعدی بررسی می‌کنیم. سیگنال از دو فیلتر بالاگذر و پایین‌گذر H و L عبور می‌کند. سپس با ضریب دو down-sample می‌شود.



شکل ۶: بلوک دیاگرام یک کدر معمولی تصویر بر اساس موجک.



شکل ۷: فیلتربانک تجزیه و بازسازی.



شکل ۸: **DWT** دوبعدی اعمال‌شده بر روی سطرها و ستون‌های تصویر.

تبدیل معکوس نیز با up-sample کردن و سپس استفاده از فیلترهای بازسازی L' و H' می‌باشد که در اکثر مواقع معکوس فیلترهای L و H می‌باشند (شکل ۷). دو فیلتر تجزیه CDF(۲،۲) عبارتند از [۱۰]

$$H = d_{1,i} = s_{.,2i+1} - [\frac{1}{2}(s_{.,2i} + s_{.,2i+2}) + \frac{1}{2}] \qquad (11)$$

$$L = s_{1,i} = s_{.,2i} + [\frac{1}{4}(d_{1,i-1} + d_{1,i}) + \frac{1}{2}]$$

این فیلتربانک از H و L برای پیش‌بینی و به‌روز رسانی سیگنال استفاده می‌کند.

بعد از اعمال DWT یک‌بعدی روی سطرهای ماتریس تصویر اصلی، روی تصویر تبدیل‌شده نیز دوباره تبدیل DWT روی ستون‌های ماتریس صورت می‌گیرد (شکل ۸).

مربع بالایی درسمت چپ شامل تمامی ضرایبی است که توسط اعمال فیلتر پایین‌گذر L روی سطرها و ستون‌ها بدست آمده است. در اغلب مواقع، اطلاعات فرکانس بالا در تصویر از نظر چشم قابل ملاحظه نیستند پس می‌توان آنها را حذف کرد.

پیاده‌سازی الگوریتم موجک CDF(۲،۲) در نرم‌افزار MATLAB و همچنین بوسیله [۱۱] انجام پذیرفته و نتایج آن بر روی یک تصویر ۲۵۶×۲۵۶ در شکل ۹ و جدول ۵ نشان داده شده است. نسبت فشرده‌سازی ۴:۱ است.

در جدول ۶ مقایسه این چهار روش با هم و در مقایسه با روش مرجع JPEG (DCT) آورده شده است.

۵-۱ کوانتیزاسیون و کدینگ آنتروپی

هم‌اکنون تمامی زیرباندهای[1] مولفه‌های تبدیل‌شده کوانتیزه می‌شوند.

1. Sub-Band

(ب) (الف)

شکل ۹: (الف) تصویر اصلی و (ب) تصویر فشرده‌شده توسط CDF(۲،۲).

جدول ۵: نتایج پیاده‌سازی الگوریتم موجک CDF(۲،۲)

زمان اجرای فشرده‌سازی در MATLAB	۲۳٫۲۹۳۰ s
زمان اجرای فشرده‌سازی در C++	۲٫۱ s
خطای MSE	۲۹٫۴۵۹
PSNR برحسب d*B*	۳۸٫۷۴۵۹
نرخ بیت (bits/pixel)	۱

جدول ۶: مقایسه روش‌های مختلف فشرده‌سازی

	Wavelet CDF(۲،۲)	JPEG (DCT)	SVD	اهرام گاوسی	BTC
زمان اجرای فشرده‌سازی (sec) در نرم‌افزار MATLAB	۲۳٫۲۹۳۰	۲۴٫۹۴۶۰	۸۳٫۹۱۱۰	۱۱٫۰۳۶۰	۱۴٫۲۶۰۰
MSE	۲۹٫۴۵۹	۳۳٫۵۶۷	۵۲٫۹۱۱	۱۰۸٫۰۲۱۴	۳۴٫۹۹۰
PSNR برحسب dB	۳۸٫۷۴۵۹	۳۶٫۹۷۷۱	۳۰٫۶۵۸۱	۲۶٫۸۲۳۶	۳۲٫۴۶۶۵
نرخ بیت (bits/pixel)	۲	۱٫۲۵	۲	۴	۲
پیاده‌سازی سخت‌افزاری	ساده	متوسط	سخت	نسبتاً سخت	نسبتاً سخت

کوانتیزاسیون عامل اصلی فشرده‌سازی با تلفات است. تنها یک مرحله کوانتیزاسیون (یعنی عددی که ضرایب موجک به آن تقسیم می‌شوند) برای هر زیرباند مجاز می‌باشد و نتیجه به عدد صحیح کمتر قبلی گرد می‌شود [۱۲]. این کار توسط یک کد در نرم‌افزار MATLAB صورت پذیرفته است. آنتروپی مفهوم مشترکی در بسیاری از کاربردها خصوصاً پردازش سیگنال دارد. ما به کمک یک کد نوشته‌شده در MATLAB آنتروپی Shannon را روی تصویر کوانتیزه‌شده اعمال نمودیم. در عبارت زیر، s سیگنال و $s(i)I$ ضرایب s در مبنای ارتونرمال می‌باشد. آنتروپی شانون E طوری تعریف می‌شود که $E(0)=0$ و

$$E(s)=\sum_i E(s_i) \qquad (۱۲)$$

آنتروپی شانون (غیر نرمالیزه) به صورت زیر تعریف می‌شود:

$$E_1(s_i)=s_i^2\log(s_i^2)$$
$$E_1(s)=-\sum s_i^2\log(s_i)^2 \qquad (۱۳)$$

لازم به ذکر است که: $0\times\log(0)=0$.

۵-۲ روش تقسیم‌بندی

با توجه به اینکه در پیاده‌سازی سخت‌افزاری از XC۹۵۷۲ استفاده شده است، سودجستن از روش پارتیشن‌بندی بسیار مناسب و کارا به نظر می‌رسد [۱۳] زیرا به منظور محاسبه تبدیل دوبعدی DWT ما باید تصویر را در حافظه خارجی (RAM) ذخیره کنیم و در مورد XC۹۵۷۲ حافظه خارجی قادر نیست تمامی تصویر را که ممکن است از چند کیلوبایت الی چند مگابایت باشد در خود ذخیره کند.

در این حالت، اگر تصویر به بلوک‌های غیرهمپوشان به نام "زیرتصویرها" تقسیم شود، قابلیت ذخیره در RAM خارجی را خواهد داشت. سپس می‌توانیم تبدیل موجک دوبعدی را روی زیرتصویرها انجام دهیم. در اینجا سایز زیرتصویرها ۳۲×۳۲ می‌باشد. شکل ۱۰ نشان‌دهنده روش تقسیم‌بندی در مقایسه با روش معمولی است.

۶- معماری سخت‌افزار و جزئیات طراحی

طبیعت پیکربندی مجدد CPLD امکان پیاده‌سازی الگوریتم تبدیل موجک گسسته را به منظور فشرده‌سازی تصویر فراهم می‌آورد. XC۹۵۷۲ یک CPLD کارآمد است که از قابلیت برنامه‌ریزی به صورت درون-سیستم[1] و همچنین توانایی تست برای یکپارچه‌سازی منطقی برخوردار است. این CPLD از ۱۸V۳۶ بلوک تابع تشکیل شده است که ۱۶۰۰ گیت با تاخیر انتشار پایین و توانایی I/O ی ۳٫۳ یا ۵ ولت را دارد. این CPLD دارای ۷۲ ماکروسل است که هریک را می‌توان با کابل JTAG ByteBlaster توسط درگاه موازی پیکربندی کرد [۱۴].

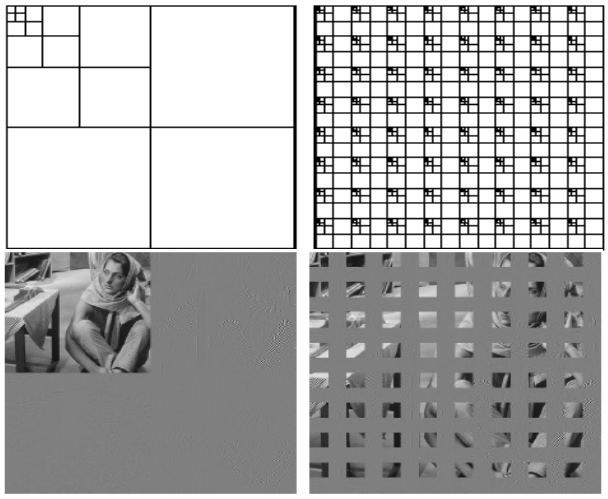
طراحی سخت‌افزاری توسط زبان VHDL نوشته‌شده و در نرم‌افزار Foundation[2] ۵٫۱ مورد سنتز قرار گرفته و سپس توسط کابل JTAG در نرم‌افزار IMPACT به روی CPLD پیکربندی شده است. تصویر اصلی موجود در کامپیوتر به زیرتصویرهایی تقسیم شده که هم بوسیله درگاه سری (توسط ATmega۸۵۳۵) و هم بوسیله درگاه موازی به CPLD قابل انتقال می‌باشد. پس از اعمال تبدیل موجک، ضرایب محاسبه‌شده به کامپیوتر ارسال می‌گردد.

۷- نتایج تجربی

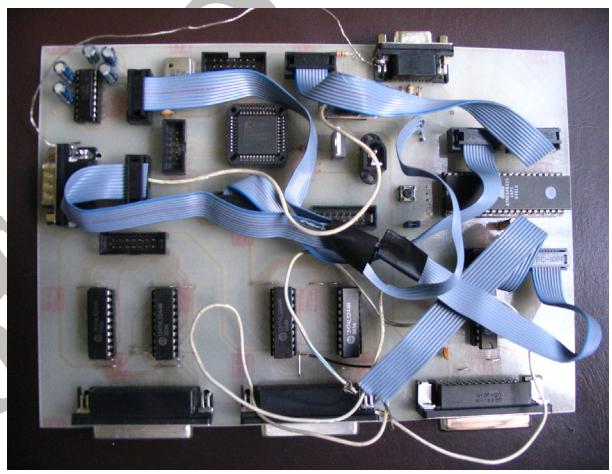
در این بررسی، زمان کلی I/O از نظر تجربی بسیار بیشتر از زمان محاسبات بوده و در نتیجه نرخ ارسال داده بین حافظه کامپیوتر پنتیوم III و CPLD زمان بسیاری را به خود اختصاص می‌دهد.

پیاده‌سازی نرم‌افزاری CDF(۲،۲) توسط یک کد در نرم‌افزار MATLAB صورت پذیرفته است. همچنین بوسیله کد C++ مقایسه‌ای میان سرعت اجرا در نرم‌افزار و سخت‌افزار صورت گرفته است. این مقایسه حاکی از ان است که پیاده‌سازی سخت‌افزاری سرعت بالاتری دارد، زیرا نرم‌افزار در ریزپردازنده کامپیوتر به صورت ترتیبی عملیات پردازش را انجام می‌دهد، درحالیکه سخت‌افزار به صورت موازی و همزمان، اطلاعات

1. In-System
2. http://www.xilinx.com



شکل ۱۰: روش پارتیشن‌بندی در مقایسه با روش معمولی.



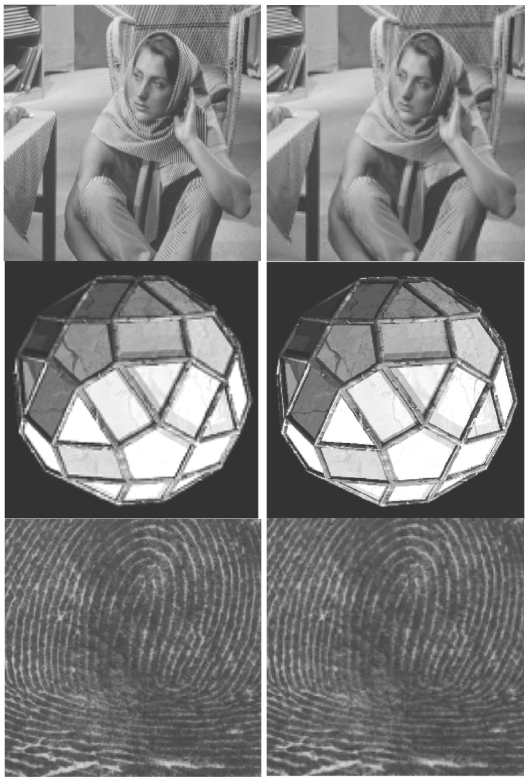
شکل ۱۱: مدار چاپی فشرده‌ساز CDF(۲،۲).

را پردازش می‌کند و فرکانس ساعت کمتری را برای رسیدن به نتیجه نیاز خواهد داشت (جدول ۷).

این طراحی در ۴۸ ماکروسل از XC۹۵۷۲ پیاده‌سازی شد که متناظر با ۶۷ ٪ از کل فضای تراشه می‌باشد. شکل ۱۱ شمایی از طرح سخت‌افزار طراحی‌شده به همراه مدارات جانبی آن را نشان می‌دهد.

شکل ۱۲ همراه با جدول ۸، نتایج بدست‌آمده را برای سه تصویر Barbara، Facets و Fingerprint نشان می‌دهد. به علت اینکه ضرایب فیلتر نیز در الگوریتم کد می‌شوند، از فیلترهای یکسانی برای فشرده‌سازی انواع تصاویر از قبیل تصویرهای طبیعی، سنتتیک، پزشکی، هوایی، اسکن‌شده، ترکیبی و ... استفاده می‌شود. این موضوع در فشرده‌سازی بدون تلفات نیز مصداق دارد. باید توجه داشت که روش کلی انتخاب فیلتر همواره بهترین کیفیت را از نقطه‌نظر کاربردهای خاص به دنبال نخواهد داشت.

متاسفانه در اکثر نتایج منتشرشده در تجربیات کدینگ تصویر از چند فیلتر خاص موجک معلوم استفاده می‌شود و همچنین تصاویر تست‌شده نیز عموماً مشابه هستند. تاکنون هیچ مطالعه سیستماتیکی برای اینکه موجک‌های متفاوت چه تاثیری بر روی کارایی و کدینگ تصاویر متفاوت دارند، صورت نپذیرفته است. بنابراین در آینده طراحی‌هایی که بر اساس CPLD صورت می‌گیرند باید از کدهای نرم‌افزاری GUI بهره گیرند که دسترسی به پایگاه اطلاعاتی موجک‌ها دارند و بطور هوشمند به ارائه طرحهای سخت‌افزاری مطابق با نیازمندی‌های کاربر از قبیل نرخ بیت، کلاس موج استفاده‌شده و کدینگ نواحی مورد نظر می‌پردازند.



شکل ۱۲: تصاویر اصلی (چپ) و تصاویر فشرده‌شده (راست).

جدول ۷: مقایسه سرعت سخت‌افزار و نرم‌افزار در روش فشرده‌سازی CDF(۲،۲)

Image	C++	MATLAB	CPLD (XC۹۵۷۲)
Barbara	۲ s	۲۱٫۲ s	۲۰۳٫۱ ms
Facets	۲٫۳ s	۲۵٫۲ s	۲۰۴ ms
Fingerprint	۱٫۹ s	۲۰ s	۱۹۵٫۳ ms

جدول ۸: نتایج روش فشرده‌سازی CDF(۲،۲)

Images	Barbara	Facets	Fingerprint
Size	۲۵۶×۲۵۶	۲۵۶×۲۵۶	۲۹۶×۲۹۶
Decomposition Level	۱	۱	۱
MSE	۲۰۴٫۷۷۶	۴۴٫۶۷۴۵	۶۴٫۴۳۲
PSNR (dB)	۲۵٫۰۱۷	۳۰٫۸۹۴	۳۰٫۰۳۸
Bit Rate(bit/pixel)	۰٫۹۰۶	۰٫۸۷۸	۰٫۹۱۶

۸- نتیجه‌گیری

بررسی شبیه‌سازی این پنج روش نشان‌دهنده این واقعیت می‌باشد که از نقطه نظر فشرده‌سازی با تلفات، الگوریتم‌های CDF(۲،۲) از سایر الگوریتم‌ها کارامدتر می‌باشند و با توجه به جدول ۶ با نرخ بیت کمتر، قابلیت فشرده‌سازی تصاویر با کیفیت بالا را دارند. لازم به ذکر است که زمان اجرای الگوریتم‌ها در نرم‌افزار MATLAB و C++ به الگوریتم برنامه‌نویسی، قدرت پردازشگر و دفعات ارجاع به حافظه بستگی دارد. همچنین بسته به نوع کوانتیزاسیون و سایز بلوک‌ها در الگوریتم‌های JPEG و موجک CDF(۲،۲) می‌توان نرخ بیت متناسب با کاربرد خاص را برآورده ساخت.

مراجع

[1] K. Z. Bukhari, *Visual Data Transforms Comparison*, M. S. Thesis, Delft University of Technology, The Netherlands, pp. 28-32, Aug. 2002.

[2] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Geometric methods for wavelet-based image compression," *Wavelets X in SPIE International Symposium on Optical Science and Technology*, pp. 507-520, San Diego, California, Aug. 2003.

[3] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Geometric tools for image compression," in *Proc. 36th Asilomar Conf. on Signals, Systems, Computers, Pacific Grove, and CA*, vol. 2, pp. 1725-1729, Nov. 2002.

[4] B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of JPEG2000," *IEEE Signal Processing. Magazine*, vol. 18, no. 5, pp. 22-35, Sep. 2001.

[5] R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, "Lossless image compression using integer to integer wavelet transforms", in *Proc. Int. Conf. on Image Proc. (ICIP)*, vol. I, pp. 596-599, 1997.

[6] M. Sima, S. Cotafona, S. Vassiliadis, and J. T. J. van Eindhoven, "8×8 IDCT implementation on an FPGA-augmented trimedia," in *Proc. IEEE Symp. on FPGAs for Custom Computing Machines FCCM 2001*, pp. 160-169, California, Apr. 2001.

[7] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. on Communications*, vol. 27, no. 9, pp. 329-336, Sep. 1979.

[8] A. Prakash Asirvatham, *Gaussian and Laplacian Pyramids*, Technical Report, International Institute of Information Technology, 2002. http://gdit.iiit.net/~arul/report/node12.html

[9] D. Kalman, "A singularly valuable decomposition: the SVD of a matrix," *The College Mathematics J.*, vol. 27, no. 1, pp. 2-23, Jan. 1996.

[10] J. Shapiro, "Embedded image coding using zero trees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[11] S. G. Mathen, *Wavelet Transform Based Adaptive Image Compression on FPGA*, M. S. Thesis, University of Calicut, Calicut, India, 1995.

[12] D. Rebollo and B. Girod, "Design of optimal quantizers for distributed source coding," in *Proc. Data Compression Conf. DCC*'03, pp. 13-22, Mar. 2003.

[13] J. Ritter and P. Molitor, "A pipelined architecture for partitioned DWT based lossy image compression using FPGA's," in *Proc. IEEE Conf. FPGA*, pp. 201-206, 2001.

[14] *XC9572 In-System Programmable CPLD*, Xilinx's product specification datasheet, Version 3.0, released Dec. 4, 1998.

عباس‌علی لطفی نیستانک متولد ۱۳۵۰ در تهران بوده و تحصیلات خود را در مقاطع کارشناسی مخابرات و کارشناسی ارشد الکترونیک بترتیب در سالهای ۱۳۷۲ و ۱۳۷۶ و دکتری مخابرات در سال ۱۳۸۳ از دانشگاه علم و صنعت ایران به پایان رسانده است. نامبرده در سالهای ۱۳۷۴ الی ۱۳۷۷ به عنوان مهندس فنی در طراحی شبکه در صدا و سیمای جمهوری اسلامی ایران مشغول بکار بوده است. ایشان در چند سال اخیر در دانشگاه‌های آزاد اسلامی، علم و صنعت و شاهد و پژوهشکده برق جهاد دانشگاهی به تدریس و تحقیق اشتغال داشته‌است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: مدارهای فعال و غیر فعال مایکروویو، آنتن‌های مایکرواستریپ، الگوریتم‌های بهینه‌سازی، طراحی شبکه‌های رادیویی و طراحی مدارهای الکترونیکی فرکانس بالا. دکتر لطفی تاکنون بیش از ۳۰ مقاله در مجلات و همایش‌های معتبر داخلی و خارجی منتشر کرده است.

محمد محقق حضرتی متولد ۱۳۶۱ تحصیلات خود را در مقطع کارشناسی در رشته مهندسی برق با گرایش مخابرات در سال ۱۳۸۳ در دانشگاه آزاد اسلامی به پایان رسانده است. آقای حضرتی تاکنون ۳ مقاله در مجلات و همایش‌های داخلی و خارجی ارائه کرده‌اند. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: فشرده‌سازی تصویر، طراحی مدارهای FPGA، میکروکنترلرهای AVR.

مهدی شارعی متولد ۱۳۶۰ در تهران بوده و تحصیلات خود را در مقطع کارشناسی مهندسی برق در سال ۱۳۸۴ از دانشگاه آزاد اسلامی به پایان رسانده است.آقای شارعی تاکنون ۳ مقاله در مجلات و همایش‌های داخلی و خارجی ارائه کرده‌اند.زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: فشرده‌سازی تصویر، طراحی مدارهای FPGA، میکروکنترلرهای AVR، کاربرد فیبرهای نوری و رادیودیتا.

نرگس احمیدی متولد ۱۳۵۷ در تهران بوده و تحصیلات خود را در مقاطع کارشناسی مهندسی کامپیوتر نرم‌افزار و کارشناسی ارشد مهندسی کامپیوتر گرایش هوش مصنوعی و رباتیک به ترتیب در سالهای ۱۳۸۰ و ۱۳۸۲ در دانشگاه صنعتی امیرکبیر به پایان رسانده است. زمینه‌های تحقیقاتی مورد علاقه ایشان پردازش تصویر، شناسایی الگو، امنیت سیستم‌های چندرسانه‌ای، مهندسی نرم‌افزار و مدیریت شبکه‌های کامپیوتری می‌باشد. در سال ۱۳۸۰ ایشان برنده جایزه جوان جشنواره خوارزمی گردید و تاکنون ۹ مقاله در کنفرانس‌های معتبر داخلی و خارجی ارائه کرده‌اند.