پنجمین کنفرانس سیستمهای فازی ایران
هفدهم الی نوزدهم شهریورماه ۱۳۸۳

طراحی یک نرم افزار برای فشرده‌سازی تصاویر دیجیتال با استفاده از کنترلر فازی

جواد عبدی، کارو لوکس[۱]، اعظم فامیل خلیلی [۲]

مؤسسه استاندارد و تحقیقات صنعتی ایران ـ شرکت کیاتل (طرح و تجهیز مراکز مخابراتی)

j.abdi@ece.ut.ac.ir

چکیده

تصاویر دیجیتالی دارای حجم زیادی بوده و به همین دلیل به پهنای باند مخابراتی زیادی نیاز دارند و به همین علت فشرده سازی تصویر از اهمیت ویژه‌ای در پردازش تصاویر برخوردار است. روش‌های فشرده‌سازی به دو دسته تقسیم می‌شوند: روش‌های بدون خطا یا برگشت‌پذیر و روش‌های همراه با خطا یا غیرقابل برگشت. در روش‌های برگشت‌پذیر تصویر فشرده شده دقیقاً بازرسی می‌شود ولی نرخ فشرده‌سازی آن پایین می‌باشد اما روش‌های همراه با خطا دارای نرخ فشرده‌سازی بالایی می‌باشد

واژه‌های کلیدی: دسته‌های فازی - روش استنتاج ماکزیمم - روش Run_Length

مقدمه

در مقاله فوق سعی شده است تا با بکار بردن نظریه فازی روش ساده‌ای برای فشرده‌سازی تصاویر دیجیتال - که امروزه کاربرد فراوانی پیدا کرده‌اند - ارائه گردد. در بخش اول مقاله، روش‌های فشرده‌سازی تصاویر (دیجیتال و غیر دیجیتال) بطور کامل بیان شده است. در قسمت دوم مقاله به ساختار اصلی برنامه و بلوک دیاگرام آن اشاره شده است. در بلوک دیاگرام برنامه سعی شده تا با استفاده از زبان برنامه نویسی Borland C++ برنامه‌ای ارائه شود تا بتواند تمامی مباحث فازی را بصورت گسسته پیاده سازی کند. در بخش سوم از این مقاله به فشرده‌سازی تصاویر دیجیتال با استفاده از سیستم‌های فازی پرداخته شده است و در بخش چهارم از مقاله به توابع بکار برده شده اشاره‌ای شده است و بالاخره در آخر نتایج عملی سیستم فشرده ساز را مشاهده می‌کنید.

روش‌های فشرده‌سازی

روش فشرده‌سازی Run_Length

یکی از ساده‌ترین روش‌ها، کدگذاری به طریقه Run_Length است. در این روش، یک سری از مقادیر تکراری (برای مثال، مقادیر پیکسلی) را با یک مقدار و یک شمارنده تعداد جایگزین می‌شود. مثلاً اگر از حروف برای نشان دادن اعداد و

۱- استاد، گروه مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران، مرکز تحقیقات فیزیک نظری ایران IPM ، lucas@ipm.ir
۲- مهندس طراح، شرکت کیاتل (طرح و تجهیز مراکز مخابراتی) ـ a_khalil59@yahoo.com

و مقادیر استفاده کرده باشیم، یک سری از مقادیر bbbbbbbccddddeeddd با 1a7b2c4d2e3d جایگزین خواهد شد. این روش برای پیاده سازی آسان است و در جایی که رشته‌های طولانی از مقادیر تکرار شونده وجود دارد مفید است. تصاویری که دارای نواحی بزرگ سایه‌دار و با یک رنگ باشند مثل آنچه در برنامهٔ نقاشی "Paint" تولید می‌شود، کاربردهای خوبی برای این نوع فشرده‌سازی محسوب می‌شوند. از جمله پیاده‌سازی‌های پر طرفدار روش فشرده‌سازی Run_Length، نرم‌افزار "PackBits" نام دارد که برای داده‌های نقش بیتی در کامپیوترهای Apple Macintosh بکار می‌رود.

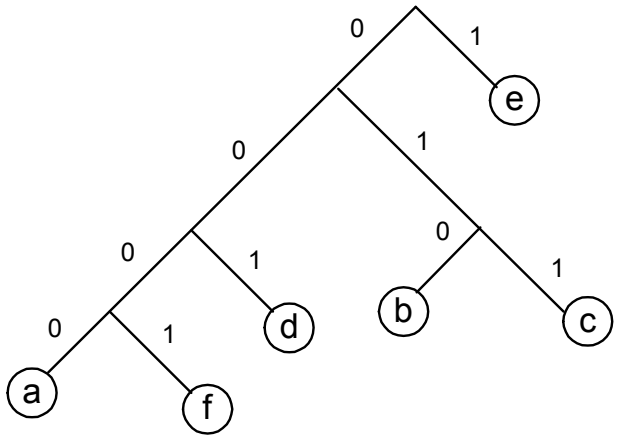
نرم افزار PackBits با فرض آنکه داده‌ها ۸ بیتی‌اند، مقادیر تکرار شونده را به دو صورت دو بایتی کدگذاری می‌کند. اولین بایت شامل یک عدد است، یعنی n که بین ۱۲۷- و ۱- بوده و شماره تکرار یعنی n+1- را تولید می‌نماید. بایت دوم شامل مقدار تکرار شونده است. مقادیر غیر تکراری مثل abcde، با یک کد شروع شونده یک بایتی، m بین ۰ و ۱۲۷ مورد کدگذاری قرار می‌گیرند که در آن m+1 (m در این مثال ۴ است) طول رشته خواهد بود. طول مورد نظر نمی‌تواند بیش از ۱۲۸ بایت باشد و در غیر اینصورت باید شکسته شود. به صورت معمول، فشرده‌سازی از یک خط پویش به دیگری ادامه نمی‌یابد. کدگذاری Run_Length در بسیاری از قالب‌های فایلی نقش بیتی توضیح داده شده مورد استفاده قرار می‌گیرد، مثل: MacPaint , TIFF , GEM , PCX.

روش کدگذاری Huffman

کدگذاری Huffman (1952) یک روش معمول برای فشرده‌سازی است. اساس کار آن، در نظر گرفتن کدهای معمولتر برای داده‌هاست. در روش فوق که در سال ۱۹۵۲ برای فایل‌های متنی ابداع شد، گوناگونی‌های زیادی وجود داشت. شمای ابتدایی، واگذاری یک کد دودویی به هر مقدار واحد، با کدهایی که طول متفاوت دارند، صورت می‌گیرد. این واگذاری مقداری، در یک جدول تبدیل ذخیره می‌شوند که این جدول قبل از خود کدها در اختیار نرم افزار از کد درآورنده قرار می‌گیرد. برای مثال، ۶ مقدار واحد در abbbccddeeeeeeeeef وجود دارند. فرکانس‌های تکرار به قرار زیرند:

a:1, b:3, c:4, d:2, e:9, f:1

برای ایجاد یک کد کمینه برای هر یک، از یک درخت دودویی مثل شکل (۱) استفاده می‌شود.



شکل ۱ - یک درخت دودویی برای کدگذاری Huffman

روش فشرده‌سازی LZW

LZW یک الگوریتم فشرده‌سازی جدید محسوب می‌شود که توسط Wetch طرح و توسط Lempel و ZIV در سال ۱۹۸۴ ساخته شده است. بر خلاف Huffman، LZW نیازی به تشکیل جدولی از کدها در خلال کدگذاری ندارد. از یک جدول ساده، جدولی مؤثرتر از کدها تشکیل می‌شود که حالت سازگار دارد. همچنین بر خلاف Huffman، پیاده‌سازی‌های LZW

معمولاً از کدهای کوتاهتر برای مقادیر دارای فرکانس‌های بیشتر بهره می‌برند.

روش فشرده‌سازی ریاضی

این روش، مثل کدگذاری Huffman از کدهای کوتاهتر برای چیزهای پرفرکانس‌تر و از کدهای طولانی‌تر برای چیزهای کم فرکانس‌تر استفاده می‌کند. البته این روش مثل LZW رشته‌هایی از مقادیر را فشرده می‌کند و نه فقط خود مقادیر را یعنی شمای کدگذاری آن در صورتی که داده‌ها دارای رشته‌های یکی و تکراری باشد به بهترین نحو عمل می‌کند. فشرده‌سازی ریاضی به مباحث تئوریک برای فشرده‌سازی، نزدیک است.

گوناگونی زیادی برای این فشرده‌سازی وجود دارد. بطور ساده، هر ترتیب مختلف داده‌ای به ناحیه روی یک عدد فرضی بین ۰ و ۱ تصویر می‌شود. آن ناحیه با کسری دودویی از درصد متغیر (تعداد بیت‌ها) به نمایش گذارده می‌شود. داده‌های کم تکرار به تعداد درصد بالاتر (تعداد بیت‌های بیشتر) احتیاج دارند. بعضی از منابع خوب در زمینه تئوری فشرده‌سازی ریاضی، مقاله‌های Witten(1987) , Rissanen(1979) , Langdon(1984) , Abrahamson(1982) می‌باشند.

ساختار اصلی برنامه

کنترلرهایی که تاکنون در پروژه‌های دیگر معرفی شده‌اند اکثراً صنعتی بوده و دارای فیزیک کنترلی می‌باشند و از روی خطای حاصل از تفاضل ورودی و خروجی پیروی می‌نماید ولی کنترلی که در این مقاله معرفی شده یک کنترلر حلقه باز می‌باشد و احتیاجی به هیچ فیزیک خاصی ندارد و یک طرح کاملاً ابتکارانه می‌باشد و از روش‌های بسیار آسان فازی پیروی می‌کند و نتایج حاصله نیز بسیار مطلوب و ملموس می‌باشد. چون هدف از این مقاله بیان کاربرد کنترلر فازی در فشرده‌سازی بود، با بررسی روش‌های گوناگون فشرده‌سازی تصاویر دیجیتال به نظر رسید که روش فشرده‌سازی Run_Length بدلیل انطباق با کنترلرهای فازی و بازیابی ساده تصویر مناسب باشد. زیرا یکی دیگر از کارهای انجام شده در این مقاله بازیابی تصویر اصلی از تصویر فشرده می‌باشد تا بتوان نتیجه کار را با مدل‌های گوناگون کنترل مقایسه کرد.

در نهایت به خاطر خطای چشمی، از نرم افزار دیگری که دو تصویر دیجیتال را بطور همزمان دریافت نموده و اختلاف هر سطر مشخصی از هر دو تصویر را همزمان رسم نماید، استفاده شده است. منحنی حاصل شده که منحنی هیستوگرام شدت روشنایی مربوط به همان سطر مشترک می‌باشد بیانگر میزان تغییرات حاصله از فشرده‌سازی می‌باشد.

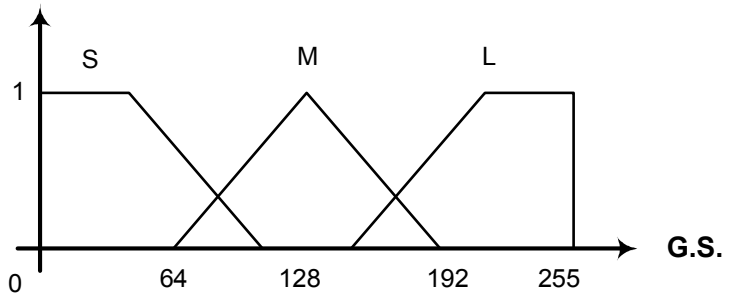
فشرده‌سازی تصویر دیجیتال توسط کنترلر فازی

در این مقاله از فایل گرافیکی BMP (نقش بیتی تحت ویندوز) برای تحلیل روش فوق استفاده شده است، زیرا اطلاعات تصویری آن به راحتی در اختیار بوده و از آن استفاده کرد. در ابتدای هر کار تصویرهای گوناگون را بر هر پسوندی، توسط نرم افزارهای گرافیکی تبدیل به یک تصویر BMP خاکستری گونه نموده تا دارای مشخصات زیر شوند. این تصویر دارای یک Header تصویر یا به عبارت بهتر مفسر تصویر برای رایانه می‌باشد که در همه تصویرهای BMP (رنگی یا خاکستری گونه) مشترک و غیر قابل تغییر می‌باشد. حجم این Header برابر ۱۰۷۸ بیت بوده و بعد از این تعداد اطلاعات اصلی مربوطه به فایل تصویری موجود می‌باشد. هر بیت از این اطلاعات ما بین (00h) تا (FFh) که معادل (0d) تا (256d) می‌باشد معرف شدت روشنایی هر پیکسل از تصویر است. در روش Run_Length هر پیکسل با پیکسل قبلی خود مقایسه و اگر شدت روشنائی‌های یکسان داشته باشند پیکسل جدید را ثبت نکرده و در عوض یک واحد به شمارنده اضافه می‌کرد این روند تا جائی ادامه داشت که پیکسل جدید با پیکسل‌های قبلی مطابقت نداشته باشد، در اینصورت ابتدا مقدار شمارنده را ذخیره کرده و سپس نماینده آن پیکسل‌های تکراری را ذخیره می‌کند. برای خواندن تصویر فشرده شده سعی شده که مقدار شمارنده باکد (-) نشان داده شود، در اینصورت در ابتدا معادل عددی کد (-) و سپس مقدار شمارنده و در نهایت مقدار نماینده پیکسل‌ها قرار داده می‌شود.این موضوع در مثال زیر نشان داده شده است.

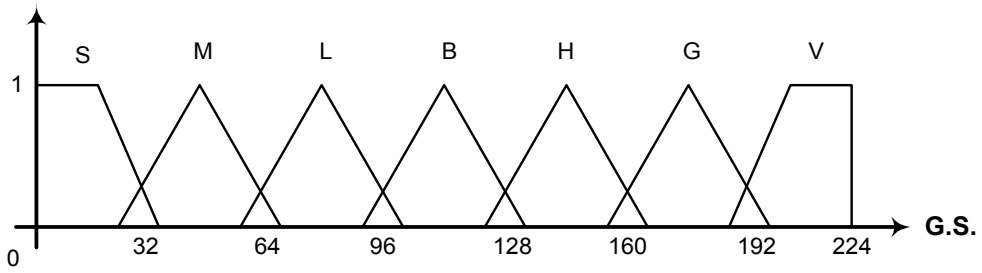
1111188888888666225

51_98_63_22_15

کنترلر فازی مورد استفاده دارای سه دسته فازی بوده و درجه عضویت‌ها نیز از مجموعه مثلثی انتخاب شده‌اند که در شکل (۲) نشان داده شده است. برای بالاتر بردن دقت عملکرد سیستم در ادامه علاوه بر سه دسته فازی از هفت دسته فازی نیز استفاده شده است، که شکل آن و همینطور نحوه تقسیم بندی پیکسل‌ها در شکل (۳) نشان داده شده است.



شکل ۲ – سه دسته فازی مثلثی برای فشرده فازی تصویر



شکل ۳ – هفت دسته فازی مثلثی برای فشرده فازی تصویر

همانطور که در شکل (۶) نشان داده شده است محدودهٔ تغییرات مجموعه‌ها، منطبق با محدودهٔ تغییرات شدت روشنایی پیکسل‌های تصویر می‌باشد. عملیات کنترلر فازی بدین نحو است که در ابتدا یکی یکی شدت روشنایی پیکسل‌ها را از فایل اصلی تصویر خوانده و سپس تک تک آنها را مورد سنجش قرار داده و مشخص می‌نماید که متعلق به چه مجموعه‌هایی هستند و همچنین در هر مجموعه دارای چه درجه عضویتی هستند. هر عضو که بتواند دارای دو درجه عضویت باشد به دسته‌ای تعلق دارد که دارای درجهٔ عضویت بیشتری نسبت به آن دسته باشد، پس به راحتی با ماکزیمم‌گیری بین دو درجه عضویت می‌توان دسته دلخواه را انتخاب نمود. در این حالت پیکسل دیگر تعلق به نقطه خاصی نداشته و فقط تعلق به محدوده خاصی دارد که محدوده دستهٔ معرفش می‌باشد. حال پیکسل بعدی که فراخوانده شد مانند قبل محدوده‌اش مشخص می‌شود. در نهایت تمام پیکسل‌ها از ۰ تا ۲۵۶ حالت برای تمایز بیشتر ندارند یعنی همان سه دسته فازی و به ازای پیکسل‌هائی که در یک مجموعه معین تکرار می‌شوند شمارنده پر می‌گردد و هرگاه پیکسلی متعلق به دسته‌ای دیگر ظاهر شد دیگر وقت آن می‌رسد که کد شمارنده همراه مقدار شمارنده و نماینده پیکسل‌ها چاپ شوند.

توابع برنامه فشرده‌سازی

اولین تابع بکار رفته تابع Max می‌باشد که یک تابع تعریف شده در نرم افزار Borland C++ می‌باشد. دومین تابع برنامه تابع MemberShip می‌باشد. این تابع هر پیکسل ورودی را که خوانده می‌شود تعیین دسته و درجه عضویت می‌نماید. سومین تابع، تابع Rule Base می‌باشد که مهمترین تابع است که دسته هر پیکسل را با دسته پیکسل بعدی مقایسه نموده و در نهایت اعلام می‌دارد که چه تصمیمی باید گرفته شود. بعنوان مثال اگر یک پیکسل متعلق به دسته S باشد و پیکسل بعدی نیز تعلق

به دسته S داشته باشد آنگاه مقدار نهائی همان دسته S می‌باشد یعنی تصمیماتی که در مورد دسته S داریم برای پیکسل دوم نیز اجرا می‌گردد. تابع بعدی، تابع Defuzzy می‌باشد که با توجه به هر دسته مقدار غیر فازی شده هر پیکسل (نماینده) را برای مقایسه با پیکسل بعدی و یا ذخیره نهائی برمی‌گرداند.

تابع آخر، تابع Main می‌باشد که ابتدا فایل‌های بخصوص را می‌خواند و سپس مقدار Header را که کنار گذاشته و تک تک پیکسل‌ها را فرا می‌خواند. سپس تابع Member Ship را فرا می‌خواند. این تابع مقدار روشنائی و درجه عضویت مخصوص هر پیکسل و در نهایت دسته مزبورش را می‌نماید. سپس تابع Rule Base فراخوانی می‌گردد تا پیکسل را با توجه به دسته مورد نظرش تجزیه و ترکیب نماید. یعنی همانطور که اشاره شد پیکسل اول خوانده ذخیره شده و پیکسل‌های دوم (بعدی) نسبت به آن سنجیده می‌شود در حالت عادی پیکسل اول را صفر معرفی کرده و پیکسل دوم را که در واقع اولین پیکسل فایل گرافیکی می‌باشد را فرا خوانده و آنرا تعیین هویت نموده و بطور مثال اگر جز، دسته مجموعه S باشد و از درجه عضویت $\mu = out(1) = 0.6$ برخوردار باشد، ابتدا با استفاده از تابع ماکزیمم‌گیری درجه عضویت ماکزیمم بین هر پیکسل را بدست آورده، سپس تابع Defuzzy را فرا خوانده تا درجه عضویت ماکزیمم را در نمایندهٔ دسته S (در این روش برابر ۶۴) ضرب نماید و آنرا که یک عدد فازی شده مربوط به پیکسل دوم (همان پیکسل اول گرافیکی) می‌باشد را برگرداند.

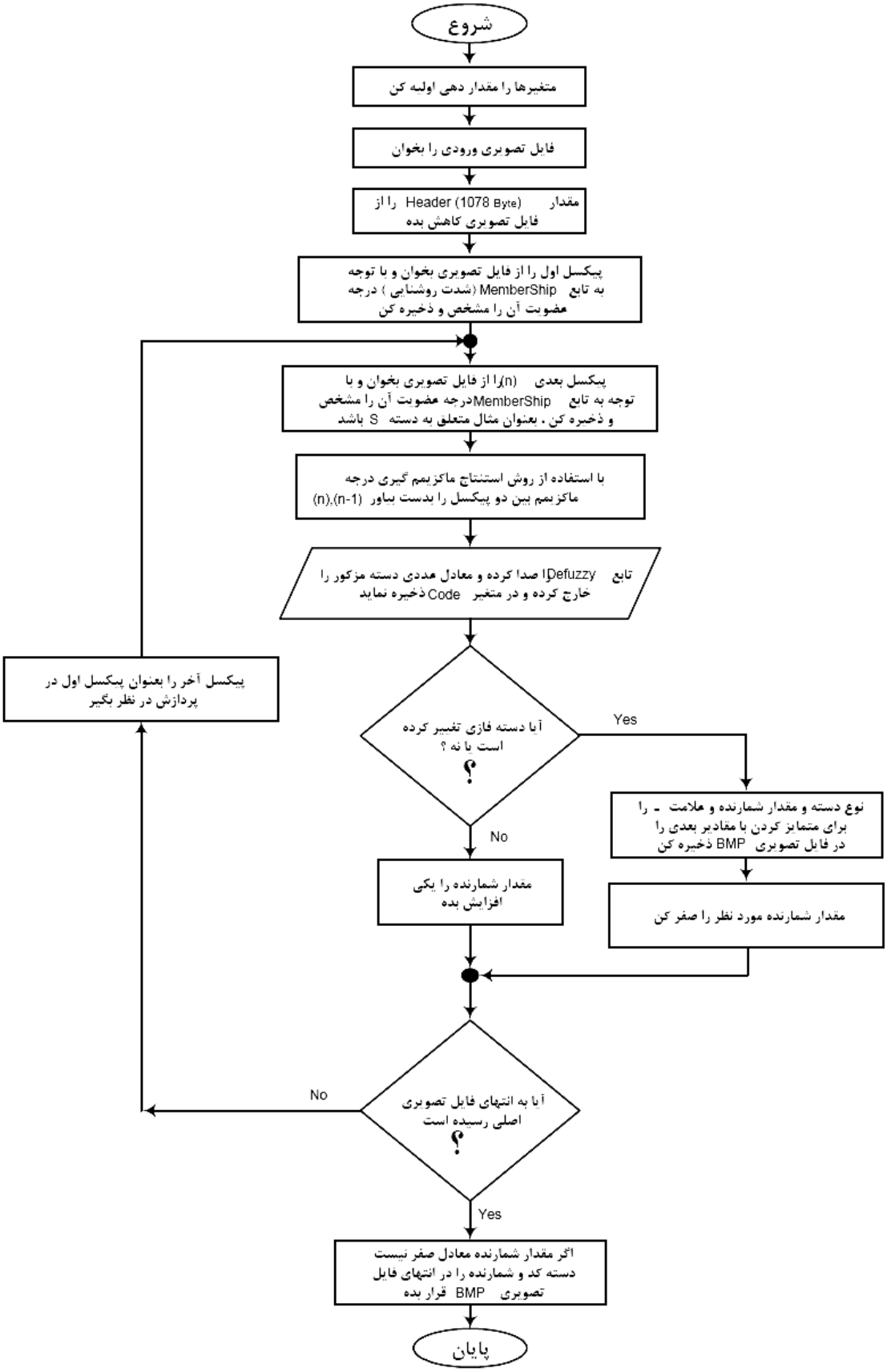
حال با توجه به دستهٔ عضویت آن عدد که مثلاً همان S است، این عدد بعنوان Code ذخیره می‌شود. سپس پیکسل دیگری فرا خوانده (پیکسل دوم فایل تصویری) و همان مراحل گذشته روی Code و پیکسل دوم انجام داده و در نهایت عددی از پیکسل دوم حاصل می‌شود که باز به دستهٔ آن اهمیت داده و اگر S بود، عدد دوم را بعنوان Code ذخیره کرده و عدد شمارنده را یک واحد افزایش داده و اگر همین روند برای پیکسل‌های بعدی تکرار شود و همگی S باشند ، آنها بسوی نماینده دسته S متمرکز می‌شوند یعنی شدت روشنائی‌هائی که در محدودهٔ نوری ۰ الی ۱۲۸ قرار گرفته‌اند اگر درجه عضویتشان به S بیشتر باشد به عدد ۶۴ گرایش پیدا می‌کنند و بدین نحو بسیاری از اطلاعات از بین خواهد رفت. برای بهبود این روش بهترین راه حل ممکنه افزایش تعداد دسته‌های فازی موجود می‌باشد و یا به عبارت دیگر تعداد رنگ‌های متمایز (نماینده هر دسته) افزایش داده بدین ترتیب با آنکه از حجم فشرده سازی تا حدی کاسته می‌شود ولی نتیجه مطلوب‌تری حاصل می‌گردد.

در پروژه‌های فشرده‌سازی انجام شده نکته بسیار مهم در انجام مطلوب کار مربوط به رنگ پردازی زمینه تصاویر بوده است، بدینصورت که هر چه رنگ زمینه با رنگ اشیا، متمایزتر باشد یعنی فاصله شدت روشنائی آنها بیشتر می‌بود در نتیجه حجم فایل فشرده شده بیشتر می‌گردید، ولی در عوض تصویر بازسازی شده دارای کیفیت بسیار مطلوب‌تری می‌بود و بالعکس یعنی هر چه رنگ زمینه تصویر و اشیا، تصویر بهم نزدیک‌تر بود حجم فایل فشرده شده بسیار کمتر و کیفیت تصویر بازسازی شده نیز بسیار کمتر می‌شد بطوریکه بعضی از لبه‌های تصویر با زمینه تصویر یکی می‌شد. بلوک دیاگرام کلی نرم‌افزار در شکل (۴) نشان داده شده است.

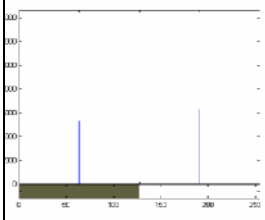
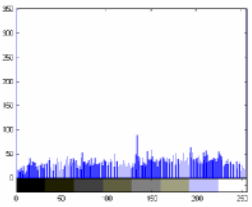
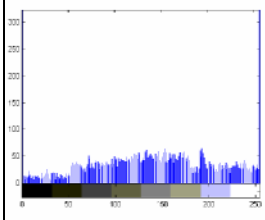
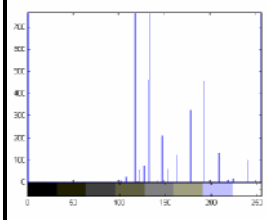
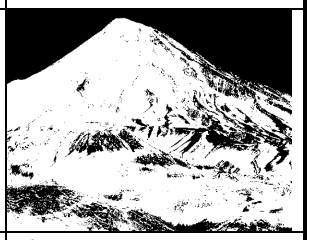
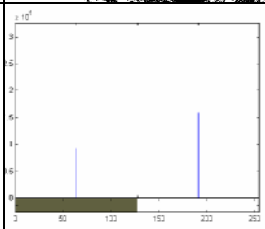
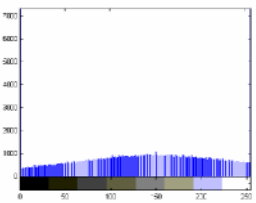
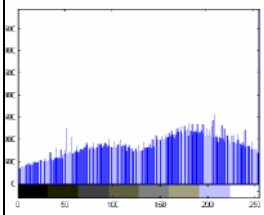
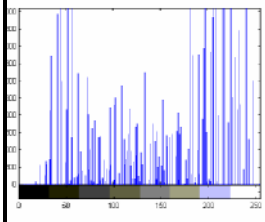
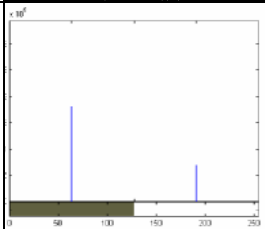
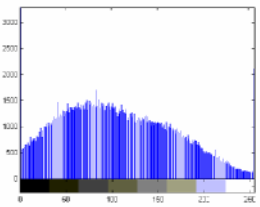
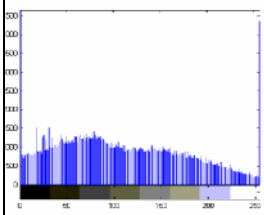
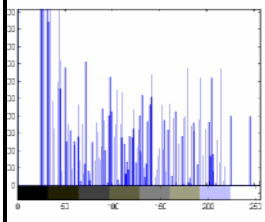
نتایج عملی

در شکل (۵) تعدای از تصاویر فشرده همراه با نمایشی از فایل اصلی، نشان داده شده است. با استفاده از یک الگوریتم ساده فشرده‌سازی و همچنین روش کنترل فازی، می‌توان حجم زیادی از اطلاعات را فشرده و یا بازسازی نمود.

انتخاب نوع دسته فازی بستگی به نوع عملکرد سیستم و دقت عملکردی سیستم دارد و همانطور که نشان داده شد می‌توان تصاویر بازسازی شده را از روی تصاویر فشرده شده بدست آورد. در ضمن ظرفیت تصاویر فشرده شده با هفت دسته فازی تفاوت زیادی با ظرفیت تصاویر اصلی ندارد. در ادامه شبیه‌سازی‌ها نمودار هیستوگرام هر تصویر نشان داده شده است. در تصاویر فشرده شده درصد پوشش خاکستری تصویر بسیار بالا رفته، یعنی اینکه با فشرده‌سازی تصویر طیف بیشتری از رنگ خاکستری در تصویر بکار برده شده است و این در حالی است که تصویر بازسازی شده تنها دارای دو نمونه از رنگ خاکستری می‌باشد.



شکل ۴ - بلوک دیاگرام کلی نرم‌افزار

تصویر بازسازی شده از روی هفت دسته فازی	تصویر فشرده‌سازی شده با سه دسته فازی	تصویر فشرده‌سازی شده با هفت دسته فازی	تصویر اصلی
			
8475 Bytes	4285 Bytes	13584 Bytes	16384 Bytes
			
			
189025 Bytes	80554 Bytes	195280 Bytes	253952 Bytes
			
165055 Bytes	68585 Bytes	184548 Bytes	248254 Bytes

شکل ۵ – نتایج شبیه‌سازی همراه با نمودارهای هیستوگرام

نتیجه گیری

از جمله کاربردهای مهم این روش در شناسائی هدف‌های متحرک برای سیستم‌های دزدگیری می‌باشد. مثلاً دوربینی که تصویری را دریافت می‌نماید، تصویر بلافاصله بازیابی می‌گردد و منتظر تصویر بعدی برای بازیابی می‌شود. وقتی که دو تصویر در اختیار باشد می‌توان برنامه‌ای طراحی کرد که اطلاعات هر دو تصویر را سریعاً با هم مقایسه نماید هر جا که تغییری در اطلاعات حاصل شده باشد چون سطر و ستون آن مشخص است، می‌توان مختصات هدف متحرک را تعیین نمود و عملیات خاص مربوطهٔ دیگر که لازم باشد را به اجرا درآورد.

مراجع

[۱] سید علی جذبی، کارو لوکس، "توسعه روش‌های یادگیری تقویتی در کنترل هوشمند و کاربردهای صنعتی و آزمایشگاهی آن"، پایان نامه کارشناسی ارشد، ۱۳۷۷

[۲] محمد تشه لب، نیما صفاپور و داریوش افیونی، "سیستم‌های فازی و کنترل فازی"، ۱۳۷۸، ناشر: دانشگاه خواجه نصیر الدین طوسی

[۳] علی غفاری، عادل مقصود پور، علیرضا پور ممتاز و جمشید قسیمی، "تفکر فازی"، ۱۳۷۷، ناشر: دانشگاه خواجه نصیر الدین طوسی

[4] C.V. Altock, B. Krause, and H.J. Zimmermann, "Advanced fuzzy logic control of a model can in extreme situation ", Fuzzy sets and system, 1995, Vol.48, No.2, P.P. 41-48.

[5] J.F. Baldwin, "Fuzzy reasoning in FIRL for fuzzy control and knowledge based application ", 1993, Asia pacific engineering, Vol.3, No.1-2, P.P. 59-82.

[6] W.Bandler, and L.J. Kohout, "Schematics of implication operators and fuzzy relational products", 1990, International of man machine studies, Vol.12, No.1, P.P. 89-116.

[7] R.E. Bellman, and M. Giertz, "On the analytic formalism of the theory fuzzy sets", 1973, Information sciences, Vol. 5, P.P. 179-156.

[8] R.E. Bellman, and L.A. Zadeh, "Decision making in a fuzzy environment", 1970, Management science, Vol.17, No. 4, P.P. 141-164.

[9] R.C. Gonzales and P. Wintz, "Digital image processing", 1987, 2nd Edition reading mass Addition-Wesley.

[10] A.K. Jain, "Fundamental of digital image processing", Prentice Hall International Editions, 1989.

[11] W.K. Pratt, "Digital image processing", New York: Wiley Interscience 1978.

[12] A. Rosenfield and A.C. Kak, "Digital image processing", Vol.1,2, New York: Academic Press, 1989.