

کتاب پردازش تصویر دیجیتال گونزالس  
[برای دانلود کتاب کامل اینجا کلیک کنید](#)

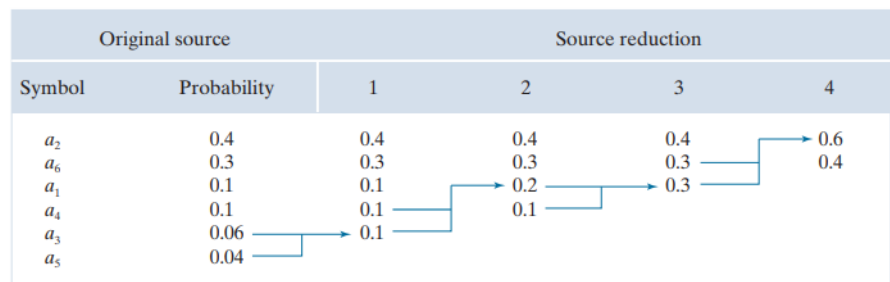
نام	سازمان	شرح
تصاویر ثابت با تن پیوسته		
BMP	Microsoft	نقشه بیت ویندوز. یک قالب فایل عمدتاً برای تصاویر ساده فشرده نشده استفاده می‌شود.
GIF	CompuServe	قالب تبادل گرافیکی یک قالب فایل که از کدگذاری LZW بدون اتلاف [۸,۵] برای تصاویر ۱ تا ۸ بیتی استفاده می‌کند. اغلب برای ساخت انیمیشن‌های کوچک و فیلم‌های کوتاه با وضوح پایین برای اینترنت استفاده می‌شود.
PDF	Adobe Systems	قالب سند قابل حمل فرمت نمایندگی اسناد دو بعدی به صورت دستگاه و وضوح مستقل. می‌تواند به عنوان ظرفی برای JPEG ، JPEG ۲۰۰۰ ، CCITT و سایر تصاویر فشرده عمل کند. برخی از نسخه‌های PDF به استانداردهای ISO تبدیل شده‌اند.
PNG	World Wide Web Consortium (W3C)	گرافیک شبکه قابل حمل فرمت فایلی که تصاویر بدون رنگ کامل را با شفافیت (حداکثر ۴۸ بیت/پیکسل) با کدگذاری تفاوت بین مقدار هر پیکسل و مقدار پیش‌بینی شده بر اساس پیکسل‌های قبلی فشرده می‌کند [۸,۱۰].
TIFF	Aldus	برچسب گذاری شده با فرمت فایل تصویری. یک فرمت فایل انعطاف پذیر از انواع استانداردهای فشرده سازی تصویر، از جمله JPEG ، JPEG-LS ، JPEG-۲۰۰۰ ، JBIG۲ و سایر موارد.
WebP	Google	WebP از فشرده سازی تلفاتی از طریق فشرده سازی ویدیویی درون فریم WebP VP۸ پشتیبانی می‌کند (زیر را ببینید) و فشرده سازی بدون اتلاف با استفاده از پیش‌بینی فضایی [۸,۱۰] و یک نوع مرجع عقب [۸,۵] و کدگذاری آنترولی هافمن [۸,۲]. شفافیت نیز پشتیبانی می‌شود.
<b>Video</b>		
AVS	MII	استاندارد صدا و تصویر. مشابه H.۲۶۴ اما از کدگذاری نمایی Golomb [۸,۳] استفاده می‌کند. توسعه یافته در چین
HDV	Company consortium	فیلم با کیفیت بالا افزونه DV برای تلویزیون HD که از فشرده سازی مشابه MPEG-۲ استفاده می‌کند، از جمله حذف افزونگی زمانی با تفاوت پیش‌بینی. [۸,۱۰]
M-JPEG	Various companies	حرکت JPEG فرمت فشرده سازی که در آن هر فریم به طور مستقل با استفاده از JPEG فشرده می‌شود.
Quick-Time	Apple Computer	محفظه رسانه ای که از DV ، H.۲۶۱ ، H.۲۶۲ ، H.۲۶۴ ، MPEG-۱ ، MPEG-۲ ، MPEG-۴ و سایر فرمت‌های فشرده سازی ویدیو پشتیبانی می‌کند
VC-1 WMV9	SMPTE Microsoft	پراکاربردترین فرمت ویدئویی در اینترنت. برای DVD های HD و Blu-ray با وضوح بالا تصویب شده است. شبیه H.۲۶۴/AVC است، با استفاده از DCT صحیح با اندازه بلوک‌های مختلف [۸,۹] و [۸,۱۰] و جداول کد متغیر وابسته به زمینه [۸,۲]، اما هیچ پیش‌بینی در فریم‌ها ندارد.
WebP VP8	Google	یک فرمت فایل بر اساس کدگذاری تبدیل بلوک [۸,۹] تفاوت‌های پیش‌بینی در فریم‌ها و بین فریم‌ها [۸,۱۰]. تفاوت‌ها با استفاده از رمزگذار حسابی تطبیقی آنترولی رمزگذاری می‌شوند. [۸,۴]

مرحله اول در رویکرد هافمن ایجاد یک سری کاهش منابع با ترتیب احتمال نمادهای مورد بررسی است ، سپس ترکیب کمترین نمادهای احتمال در یک نماد واحد که آنها را در کاهش منبع بعدی جایگزین می کند. شکل ۸.۷ این فرایند را برای کدگذاری باینری نشان می دهد (کدهای هافمن K-array نیز قابل ساخت است). در سمت چپ ، مجموعه ای فرضی از سمبل های منبع و احتمالات آنها از نظر کاهش مقادیر احتمال از بالا به پایین مرتب شده است. برای تشکیل اولین منبع کاهش ، دو احتمال پایینی ، ۰.۰۶ و ۰.۰۴ با هم ترکیب می شوند و یک "نماد مرکب" با احتمال ۰.۱ تشکیل می دهند. این نماد مرکب و احتمال مربوط به آن در ستون کاهش منبع اول قرار می گیرد تا احتمالات منبع کاهش یافته نیز از بیشترین به کمترین احتمال مرتب شود. این روند سپس تکرار می شود تا به منبع کاهش یافته با دو نماد (در سمت راست) برسید.

مرحله دوم در روش هافمن ، کدگذاری هر منبع کاهش یافته ، شروع با کوچکترین منبع و بازگشت به منبع اصلی است. البته کد باینری حداقل طول برای منبع دو نماد ، نمادهای ۰ و ۱ هستند. همانطور که در شکل ۸.۸ نشان داده شده است ، این نمادها به دو نماد سمت راست اختصاص داده شده اند. (تخصیص دلخواه است ؛ وارونه کردن ترتیب ۰ و ۱ نیز به همان خوبی کار می کند.) همانطور که نماد منبع کاهش یافته با احتمال ۰.۶ با ترکیب دو نماد در منبع کاهش یافته در سمت چپ آن ایجاد می شود ، ۰ برای کدگذاری آن استفاده می شود در حال حاضر به هر دوی این نمادها اختصاص داده شده است و یک عدد ۰ و ۱ به طور دلخواه به هر یک متصل می شوند تا آنها را از یکدیگر متمایز کند. سپس این عملیات برای هر منبع کاهش یافته تا رسیدن به منبع اصلی تکرار می شود. کد نهایی در سمت چپ در شکل ۸.۸ نشان داده شده است. طول متوسط این کد برابر است با

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5) = 2.2 \text{ bits/pixel}$$

و آنتروپی منبع ۲.۱۴ بیت/نماد است. روش هافمن کد بهینه ای را برای مجموعه ای از نمادها و احتمالات ایجاد می کند که مشروط به محدودیت این است که نمادها یک به یک کدگذاری شوند. پس از ایجاد کد ، کدگذاری و/یا رمزگشایی بدون خطا به روش جدول جستجوی ساده انجام می شود. کد به خودی خود یک کد بلاک رمزگشایی فوری است. این کد بلاک نامیده می شود زیرا هر نماد منبع در یک دنباله ثابت از نمادهای کد نقشه برداری می شود. این لحظه ای است زیرا هر کلمه کد در یک رشته از



شکل ۸.۷ کاهش منابع هافمن

Original source			Source reduction				
Symbol	Probability	Code	1	2	3	4	
$a_2$	0.4	1	0.4	1	0.4	1	0.6
$a_6$	0.3	00	0.3	00	0.3	00	0.4
$a_1$	0.1	011	0.1	011	0.2	010	0.3
$a_4$	0.1	0100	0.1	0100	0.1	011	0.1
$a_3$	0.06	01010	0.1	0101			
$a_5$	0.04	01011					

شکل ۸.۸ روش تعیین کد هافمن

نمادهای کد را می توان بدون ارجاع به نمادهای بعدی رمزگشایی کرد. این رمزگشایی منحصر به فرد است زیرا هر رشته از نمادهای کد را می توان تنها به یک روش رمزگشایی کرد. بنابراین ، هر رشته ای از نمادهای رمزگذاری شده توسط هافمن را می توان با بررسی نمادهای جداگانه رشته به روش چپ به راست رمزگشایی کرد. برای کد باینری شکل ۸.۸ ، اسکن چپ به راست رشته کدگذاری شده ۰۱۰۱۰۰۱۱۱۱۰۰ نشان می دهد که اولین کلمه کد معتبر ۰۱۰۱۰ است که کد نماد  $a_3$  است. کد معتبر بعدی ۰۱۱ است که به نماد  $a_1$  پاسخ می دهد. ادامه دادن به این ترتیب پیام کاملاً رمزگشایی شده را  $a_3 a_1 a_2 a_2 a_6$  نشان می دهد.

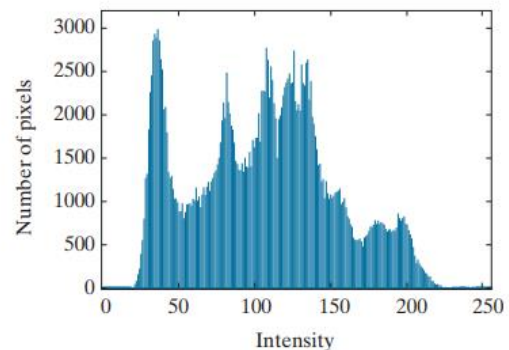
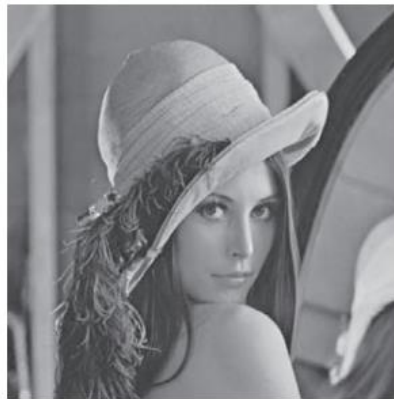
#### مثال ۸.۴: کدنویسی هافمن.

تصویر تک رنگ ۸۱۲ بیتی  $512 * 512$  در شکل ۸.۹ (الف) دارای هیستوگرام شدت نشان داده شده در شکل ۸.۹ (ب) است. از آنجا که شدت ها به یک اندازه محتمل نیستند ، از روش MATLAB روش هافمن برای کدگذاری آنها با  $7.428$  بیت/پیکسل استفاده شد ، از جمله جدول کد هافمن که برای بازسازی شدت تصویر اصلی ۸ بیتی مورد نیاز است. نمایش فشرده از آنتروپی تخمینی تصویر [۷.۳۸۳۸ بیت/پیکسل از معادله (۸-۷)] توسط  $512^2 \times (7.428 - 7.3838)$  یا  $11,587$  بیت-حدود  $0.6$  . نسبت فشرده سازی و افزونگی نسبی مربوطه به ترتیب  $C = 8/7.428 = 1.077$  و  $R = 1 - (1/1.077) = 0.0715$  است. بنابراین  $0.15$  از نمای اصلی شدت ضخامت ۸ بیت اصلی به عنوان افزونگی برنامه نویسی حذف شد. هنگامی که تعداد زیادی نماد قرار است کدگذاری شود ، ساخت یک کد بهینه هافمن یک کار بی اهمیت است. برای حالت کلی نمادهای منبع  $L$  ، احتمالات نماد  $L$  ، کاهش منبع  $L-2$  و اختصاص کد  $L-2$  مورد نیاز است. هنگامی که احتمالات نماد منبع را می توان از قبل تخمین زد ، می توان با پیش محاسبه کدگذاری "تقریباً مطلوب" را بدست آورد. کدهای هافمن چندین استاندارد متداول فشرده سازی تصویر ، از جمله استانداردهای JPEG و MPEG که در بخشهای ۸.۹ و ۸.۱۰ مورد بحث قرار گرفته است ، جداول کدگذاری پیش فرض هافمن را که بر اساس داده های تجربی از پیش محاسبه شده اند ، مشخص می کند.

#### ۸.۳ کدگذاری GOLOMB

در این بخش ، ما کدگذاری ورودی های عدد صحیح غیر منفی با توزیع های احتمالی پوسیدگی تصاعدی را در نظر می گیریم. ورودی های این نوع را می توان به صورت بهینه (به معنای قضیه اول شانون) با استفاده از خانواده ای از کدها که از نظر محاسباتی ساده تر از کدهای هافمن هستند کدگذاری کرد. خود کدها اولین بار برای نمایش طول های غیر منفی (Golomb [۱۹۶۶]) پیشنهاد شد.

با اشاره به جداول ۸.۳-۸.۵ ، کدهای Golomb در استفاده می شود • JPEG-LS • فشرده سازی AVS.



شکل ۸.۹ (الف)  $512 * 512$  A تصویر ۸ بیتی و (ب) هیستوگرام آن

در ادامه ، نماد  $\lfloor x \rfloor$  بزرگترین عدد صحیح کمتر یا مساوی  $x$  را نشان می دهد ،  $\lfloor x \rfloor$  به معنی کوچکترین عدد صحیح بزرگتر یا مساوی  $x$  است و  $x \bmod y$  باقی مانده  $x$  تقسیم بر  $y$  است. با در نظر گرفتن یک عدد صحیح غیر منفی و تقسیم کننده عدد صحیح مثبت  $m \vee 0$  ، کد Golomb از  $n$  نسبت به  $m$  ، نشان دهنده  $G_m(n)$  ، ترکیبی از کد واحد از ضرب  $\lfloor n/m \rfloor$  و نمایش باینری بقیه است.  $n * m G_m(n)$  برابر است به شرح زیر ساختار یافته است:

۱. کد واحدی ضرب  $\lfloor n/m \rfloor$  را تشکیل دهید. (کد واحد یک عدد صحیح  $q$  به صورت  $q1$  تعریف می شود و پس از آن ۰).

۲. اجازه دهید  $k = \lceil \log_2 m \rceil$  ،  $c = 2^k - m$  ،  $r = n \bmod m$  ، و باقی مانده را محاسبه کنید  $R'$ . طوری که

$$r' = \begin{cases} r \text{ truncated to } k-1 \text{ bits} & 0 \leq r < c \\ r+c \text{ truncated to } k \text{ bits} & \text{otherwise} \end{cases} \quad (8-12)$$

۳. نتایج مرحله ۱ و ۲ را به هم متصل کنید.

برای محاسبه  $G_4$  (۹) ، برای مثال ، با تعیین کد واحد از مقدار  $\lfloor 9/4 \rfloor = \lfloor 2.25 \rfloor = 2$  ، که ۱۱۰ است (نتیجه مرحله ۱) ، شروع کنید. سپس اجازه دهید  $k = \lceil \log_2 4 \rceil = 2$  ،  $r = 9$  ،  $c = 2^2 - 4 = 0$  ، که در حالت باینری  $1001 \bmod 0100$  or  $0001$  است به ترتیب دقیق با معادله (۸-۱۲) ،  $r' = 2$  (یعنی ۰۰۰۱) به ۲ بیت کوتاه می شود که ۰۱ است (نتیجه مرحله ۲). در نهایت ، ۱۱۰ را از مرحله ۱ و ۰۱ را از مرحله ۲ به هم وصل کنید تا ۱۱۰۰۱ که  $G_4$  است (۹) بدست آورید. برای مورد خاص  $k^m = 2^k$  ،  $c = 0$  and  $r' = r = n \bmod m$  بیت در

معادله (۸-۱۲) کوتاه شده است. برای همه  $n$ . تقسیمات مورد نیاز برای ایجاد کدهای Golomb حاصله به عملیات تبدیل دودویی تبدیل می شود و کدهای محاسباتی ساده تر، Rice یا Golomb-Rice هستند (رایس [۱۹۷۵]). ستون های ۲، ۳ و ۴ از جدول ۸.۶ کدهای  $G_1$ ،  $G_2$  و  $G_4$  از ده عدد صحیح غیر منفی را فهرست کرده است. از آنجا که  $m$  در هر مورد ۲ است (i.e.,  $1 = 2^0$ ,  $2 = 2^1$ , and  $4 = 2^2$ )، آنها اولین سه کد Golomb-Rice نیز هستند.

علاوه بر این،  $G_1$  کد واحد از اعداد صحیح غیر منفی است زیرا  $\lfloor n/1 \rfloor = n$  و  $n \bmod 1 = 0$  برای همه  $n$ . جدول ۸.۶ چندین کد Golomb برای اعداد صحیح ۰-۹.

$n$	$G_1(n)$	$G_2(n)$	$G_4(n)$	$G_{\text{exp}}^0(n)$
0	0	00	000	0
1	10	01	001	100
2	110	100	010	101
3	1110	101	011	11000
4	11110	1100	1000	11001
5	111110	1101	1001	11010
6	1111110	11100	1010	11011
7	11111110	11101	1011	1110000
8	111111110	111100	11000	1110001
9	1111111110	111101	11001	1110010

با در نظر داشتن اینکه کدهای Golomb فقط می توانند برای نشان دادن اعداد صحیح غیر منفی استفاده شوند و کدهای Golomb زیادی برای انتخاب وجود دارد، یک مرحله کلیدی در کاربرد موثر آنها انتخاب تقسیم کننده  $m$  است. هنگامی که اعداد صحیح ارائه شده به صورت هندسی با تابع احتمال جرم (PMF) توزیع می شوند<sup>۳۵</sup>

$$P(n) = (1 - \rho)\rho^n \quad (8-13)$$

برای برخی از  $0 < \rho < 1$ ، می توان کدهای Golomb را مطلوب نشان داد به این معنا که  $G_m(n)$  کوتاهترین طول میانگین کد را از تمام کدهای قابل رمزگشایی منحصر به فرد ارائه می دهد (Gallager and Voorhis [۱۹۷۵])، وقتی

$$m = \left\lceil \frac{\log_2(1 + \rho)}{\log_2(1/\rho)} \right\rceil \quad (8-14)$$

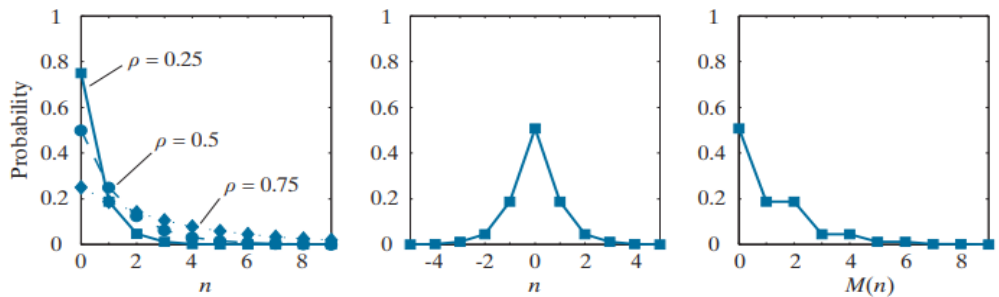
شکل ۸.۱۰ (الف) نمودارهای معادله (۸-۱۳) برای سه مقدار ۲ و به صورت گرافیکی احتمالی نمادهایی را نشان می دهد که کدهای Golomb به خوبی از آن استفاده می کنند (یعنی کد را به طور کارآمد). همانطور که در شکل نشان داده شده است، اعداد صحیح کوچک بسیار بیشتر از عدد بزرگ محتمل است. از آنجا که احتمال شدت در

<sup>۳۵</sup> یک تابع جرم احتمال (PMF) یک تابع است که این احتمال را تعریف می کند که یک متغیر تصادفی گسسته دقیقاً برابر مقداری باشد PMF. با PDF متفاوت است زیرا مقادیر PDF احتمال نیست. در عوض، انتگرال یک PDF در بازه مشخص شده یک احتمال است.

یک تصویر [برای مثال ، به نمودار او در شکل ۸.۹ (ب) مراجعه کنید] بعید است با احتمالاتی که در معادله (۸-۱۳) مشخص شده است مطابقت داشته باشد. و در شکل ۸.۱۰ (الف) نشان داده شده است ، کدهای Golomb به ندرت برای کدگذاری شدت استفاده می شوند. با این حال ، وقتی تفاوت شدت باید کدگذاری شود ، احتمالات "مقادیر تفاوت" حاصل (بخش ۸.۱۰ را ببینید) (به استثنای تفاوت‌های منفی) اغلب شبیه معادله است. (۸-۱۳) و شکل ۸.۱۰ (الف). برای رسیدگی به تفاوت های منفی در برنامه نویسی Golomb ، که فقط می تواند نشان دهنده ترکیبات غیر منفی باشد ، نقشه برداری مانند

$$M(n) = \begin{cases} 2n & n \geq 0 \\ 2|n|-1 & n < 0 \end{cases} \quad (8-15)$$

به طور معمول استفاده می شود با استفاده از این نقشه برداری ، برای مثال ، PMF دو طرفه نشان داده شده در شکل ۸.۱۰ (ب) را می توان در شکل ۸.۱۰ (ج) به PMF یک طرفه تبدیل کرد. اعداد صحیح آن دوباره مرتب شده و اعداد صحیح منفی و مثبت را متناوب می کند ، بنابراین اعداد صحیح منفی در موقعیت های صحیح مثبت عجیب و غریب ترسیم می شوند. اگر  $P(n)$  دو طرفه و مرکز باشد. در صفر ،  $P \boxtimes M(n)$  یک طرفه خواهد بود. اعداد صحیح نگاشته شده ،  $M(n)$  ، سپس می تواند کارآمد باشد. با استفاده از کد مناسب Golomb-Rice (واینبرگر و همکاران [۱۹۹۶]) رمزگذاری شده است.

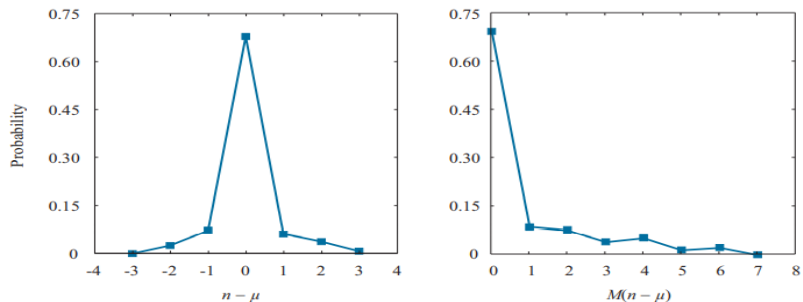


شکل ۸.۱۰ (الف) سه توزیع هندسی یک طرفه از معادله (۸-۱۳) ؛ (ب) توزیع دو طرفه در حال پوسیدگی چشمگیر ؛ و (ج) نسخه مرتب شده از (ب) با استفاده از معادله (۸-۱۵).

### مثال ۸.۵: کدگذاری Golomb-Rice

دوباره تصویر شکل ۸.۱ (ج) را در نظر بگیرید و توجه کنید که هیستوگرام آن [شکل ۸.۳ (الف) را ببینید] شبیه توزیع دو طرفه در شکل ۸.۱۰ (ب) بالا است. اگر اجازه دهیم  $n$  یک مقدار صحیح غیر منفی در باشد تصویر ، جایی که  $0 \leq n \leq 225$  و  $m$  میانگین شدت است ،  $P(n-\mu)$  توزیع دو طرفه است که در شکل ۸.۱۱ (الف). این نمودار با نرمال کردن هیستوگرام در شکل ۸.۳ (الف) با تعداد کل ایجاد شد. پیکسل در تصویر و تبدیل مقادیر نرمال شده به سمت چپ ۱۲۸ (که در واقع شدت متوسط را از تصویر کم می کند). مطابق با معادله (۸-۱۵) ،  $P(M(n-\mu))$  توزیع یک طرفه است. آسیب در شکل ۸.۱۱ (ب) نشان داده شده است. اگر مقادیر شدت

مجدد مرتب شده Golomb با استفاده از MATLAB کد G1 در ستون ۲ جدول ۸.۶ کدگذاری شود ، نمایش رمزگذاری شده ۴.۵ برابر کوچکتر از تصویر اصلی است (یعنی  $C = ۴,۵$ ). کد ۱, ۵, ۵, ۴ یا G1 یا ۸۸ of فشار تئوریک ممکن را با کدگذاری طول متغیر انجام می دهد. [بر اساس آنتروپی محاسبه شده در مثال (۸-۲) ، حداکثر نسبت فشرده سازی ممکن از طریق کدگذاری با طول متغیر  $C = 8/1.566 \approx 5.1$  است.] علاوه بر این ، کدگذاری Golomb به ۹۶ of از فشرده سازی ارائه شده توسط اجرای MATLAB دست می یابد. رویکرد هافمن ، و نیازی به محاسبه جدول کدگذاری سفارشی هافمن ندارد. حالا تصویر شکل ۸.۹ (الف) را در نظر بگیرید. اگر شدت آن Golomb با استفاده از کد G1 مشابه بالا ،  $C = ۰,۰۹۲۲$  کدگذاری شود. یعنی توسعه داده وجود دارد. این به این دلیل است که احتمال شدت تصویر در شکل ۸.۹ (الف) بسیار متفاوت از احتمالاتی است که در معادله (۸-۱۳) تعریف شده است. به طور مشابه ، کدهای هافمن می توانند در صورت رمزگذاری نمادهایی که احتمالات آنها با مواردی که کد برای آنها محاسبه شده است ، متفاوت باشد ، داده ها را گسترش دهند. در عمل ، هرچه از مفروضات احتمال ورودی که یک کد برای آن طراحی شده است فاصله بگیرید ، خطر تابع ضعیف فشرده سازی و گسترش داده ها بیشتر می شود.



شکل ۸.۱۱ (الف) توزیع احتمالی تصویر در شکل ۸.۱ (ج) پس از کسر شدت متوسط از هر پیکسل. (ب) نسخه نقشه برداری شده از (الف) با استفاده از معادله (۸-۱۵).

برای به پایان رساندن پوشش خود در مورد کدهای Golomb ، توجه داشته باشید که ستون ۵ جدول ۸.۶ شامل ۱۰ کد اول کد نمایی Golomb به ترتیب صفر ، نشان داده شده است  $\exp(n)$  کدهای نمایی - Golomb برای کدگذاری طول اجرا مفید هستند ، زیرا هر دو دور کوتاه و بلند به طور موثر رمزگذاری می شوند. دستور k-نمایی - کد  $G_k(n)$  Golomb به شرح زیر محاسبه می شود:

۱. یک عدد صحیح  $i \geq 0$  پیدا کنید به طوری که

$$\sum_{j=0}^{i-1} 2^{j+k} \leq n < \sum_{j=0}^i 2^{j+k} \quad (8-16)$$

و کد  $i$  unary را تشکیل دهید. اگر  $k = 0, i = \lfloor \log_2(n+1) \rfloor$  و کد نیز باشد. به عنوان کد گامای الیاس شناخته می شود.



۲. نمایش دوتایی از را کوتاه کنید

$$n - \sum_{j=0}^{i-1} 2^{j+k} \quad (8-17)$$

به حداقل  $k+i$  بیت‌های معنی دار

۳. نتایج مرحله ۱ و ۲ را به هم متصل کنید.

برای پیدا کردن  $G_{\text{exp}}^0(8)$  به عنوان مثال، ما اجازه می‌دهیم  $i = \lfloor \log_2 9 \rfloor$  یا ۳ در مرحله ۱ زیرا  $k = 0$ . معادله سپس

(۸-۱۶) راضی می‌شود زیرا

$$\sum_{j=0}^{3-1} 2^{j+0} \leq 8 < \sum_{j=0}^3 2^{j+0}$$

$$\sum_{j=0}^2 2^j \leq 8 < \sum_{j=0}^3 2^j$$

$$2^0 + 2^1 + 2^2 \leq 8 < 2^0 + 2^1 + 2^2 + 2^3$$

$$7 \leq 8 < 15$$

کد واحد ۳ شماره ۱۱۱۰ و معادله (۸-۱۷) است. از مرحله ۲ بازده

$$8 - \sum_{j=0}^{3-1} 2^{j+0} = 8 - \sum_{j=0}^2 2^j = 8 - (2^0 + 2^1 + 2^2) = 8 - 7 = 1 = 0001$$

که وقتی بر روی ۳ + ۰ حداقل بیت‌های مهم آن کوتاه شود ۰۰۱ می‌شود. ترکیب نتایج مرحله ۱ و ۲ ۱۱۱۰۰۰۱ را نشان می‌دهد. توجه داشته باشید که این ورودی ستون ۴ جدول ۸.۶ برای  $n-8$  است. توجه داشته باشید که مانند کدهای هافمن قسمت آخر، کدهای Golomb جدول ۸.۶ دارای طول بلوک متغیر، آنی و رمزگشایی منحصر به فرد هستند.

با ارجاع به جداول ۸.۳-۸.۵، از کدگذاری حسابی در استفاده می‌شود

• JBIG۱

• JBIG۲

• JPEG-۲۰۰۰

• H.۲۶۴

• MPEG-۴ AVC

و سایر استانداردهای فشرده سازی

#### ۸.۴ کدگذاری حسابی

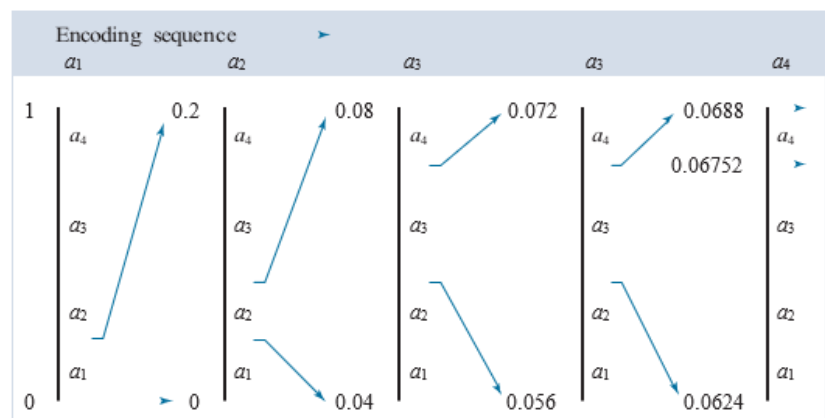
بر خلاف کدهای طول متغیر دو بخش قبلی، کدگذاری حسابی باعث ایجاد کدهای بدون انسداد می‌شود. در کدگذاری حسابی، که می‌توان از آثار الیاس (آبرامسون [۱۹۶۳]) ردیابی کرد، تناظر یک به یک بین نمادهای منبع و کلمات رمز وجود ندارد. در عوض، یک دنباله کامل از نمادهای منبع (یا پیام) به یک کلمه رمز واحد

حسابی اختصاص داده شده است. خود کلمه رمز فاصله ای از اعداد حقیقی بین ۰ تا ۱ را تعریف می کند. با افزایش تعداد نمادهای پیام ، فاصله مورد استفاده برای نمایش آن کوچکتر می شود و تعداد واحدهای اطلاعاتی (مثلاً بیت ها) برای نمایش فاصله مورد نیاز است. بزرگتر می شود هر نماد پیام متناسب با احتمال وقوع آن ، اندازه فاصله را کاهش می دهد. از آنجا که این تکنیک ، مانند رویکرد هافمن ، مستلزم این نیست که هر نماد مبدأ به تعداد انتگرال نمادهای کد تبدیل شود (یعنی نمادها در یک زمان کدگذاری شوند) ، به محدوده تعیین شده (اما فقط در تئوری) دست می یابد. توسط اولین قضیه شانون در بخش ۸.۱.

شکل ۸.۱۲ فرایند کدگذاری حسابی اساسی را نشان می دهد. در اینجا ، یک دنباله یا پیام پنج نمادی ،  $a_1 a_2 a_3 a_4$  ، از منبع چهار نماد ، کدگذاری شده است. در آغاز فرایند کدگذاری ، فرض بر این است که پیام کل بازه نیمه باز را اشغال می کند [۰ ، ۱]. همانطور که جدول ۸.۷ نشان می دهد ، این فاصله ابتدا بر اساس احتمالات هر نماد منبع به چهار ناحیه تقسیم می شود. به عنوان مثال ، نماد  $a_1$  با زیر فاصله [۰ ، ۰.۲] همراه است. از آنجا که این اولین نماد پیام در حال کدگذاری است ، فاصله پیام در ابتدا به [۰ ، ۰.۲] محدود می شود. بنابراین ، در شکل ۸.۱۲ ، [۰ ، ۰.۲] تا ارتفاع کامل شکل گسترش یافته و نقاط انتهایی آن با مقادیر محدوده محدود شده برچسب گذاری شده است. سپس محدوده محدود شده مطابق با اصل تقسیم می شود

جدول ۸.۷ مثال کدگذاری حسابی.

نماد منبع	احتمال	زیر نقطه اولیه
$a_1$	0.2	[0.0, 0.2)
$a_2$	0.2	[0.2, 0.4)
$a_3$	0.4	[0.4, 0.8)
$a_4$	0.2	[0.8, 1.0)



شکل ۸.۱۲ روش کدگذاری حسابی.

احتمال نماد منبع ، و روند با نماد پیام بعدی ادامه می یابد. به این ترتیب ، نماد  $a_2$  فاصله زیر را به [۰.۰۴ ، ۰.۰۸] ،  $a_3$  آن را به [۰.۰۵۶ ، ۰.۰۷۲] و غیره باریک می کند. نماد پیام نهایی ، که باید به عنوان نشانگر ویژه

انتهای پیام رزرو شود ، محدوده را به [۰.۰۶۷۵۲ ، ۰.۰۶۸۸] محدود می کند. البته ، هر عددی در این زیر بازه ، به عنوان مثال ، ۰.۰۶۸ ، می تواند برای نمایش پیام استفاده شود. در پیام رمزگذاری شده شکل ۸.۱۲ ، از سه رقم اعشاری برای نشان دادن پیام پنج علامت استفاده شده است. این عدد به ۰.۶ رقم اعشاری در نماد منبع تبدیل می شود و با آنتروپی منبع ، که از معادله مطلوب است ، مطلوب است. ۸.۶ ، ۰.۵۸ رقم اعشاری در هر نماد منبع است. با افزایش طول دنباله ای که کدگذاری می شود ، کد حسابی حاصله به محدوده تعیین شده توسط اولین قضیه شانون نزدیک می شود. در عمل ، دو عامل باعث می شود تابع کد نویسی از حد مجاز خارج شود: (۱) افزودن نشانگر انتهای پیام که برای جداسازی یک پیام از پیام دیگر لازم است ، و (۲) استفاده از حساب دقیق دقیق. پیاده سازی های عملی کدگذاری حسابی با معرفی یک استراتژی مقیاس بندی و یک استراتژی گرد کردن ، مشکل اخیر را برطرف می کند (لانگدون و ریسانن [۱۹۸۱]). استراتژی مقیاس بندی قبل از تسلیم آن مطابق احتمالات نماد ، هر زیر فاصله را به محدوده [۰ ، ۱] تبدیل می دهد. استراتژی گردآوری این ضمانت را می دهد که برش های مرتبط با حساب دقیق محدود مانع از نمایش دقیق زیر فصل های کدگذاری نمی شود.

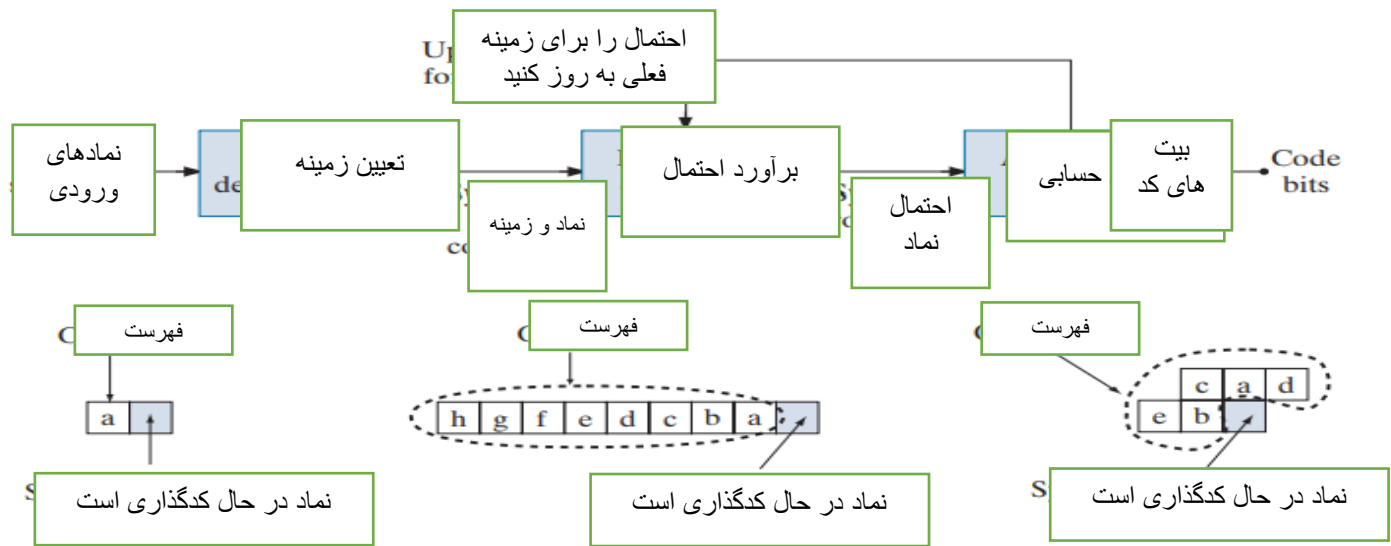
### محتوای تطبیقی برآورد احتمال وابستگی

با مدل های احتمالی نماد ورودی دقیق ، یعنی مدل هایی که احتمالات حقیقی نمادهای کد شده را ارائه می دهند ، رمزگذارهای حسابی تقریباً بهینه هستند به این معنا که تعداد متوسط نمادهای کد مورد نیاز برای نشان دادن نمادهای کد شده را به حداقل برسانند. با این حال ، مانند برنامه نویسی هافمن و Golomb ، مدل های احتمالی نادرست می توانند منجر به نتایج غیر مطلوب شوند. یک راه ساده برای بهبود دقت احتمالی استفاده از یک مدل احتمالی تطبیقی ، وابسته به زمینه است. مدل های احتمالی تطبیقی احتمالات نماد را با کدگذاری یا شناخته شدن نمادها به روز می کنند. بنابراین ، احتمالات با آمار محلی نمادهایی که کدگذاری می شوند تطبیق می یابد. مدل های وابسته به زمینه احتمالاتی را ارائه می دهند که بر اساس همسایگی از پیش تعریف شده ای از پیکسل ها ، به نام زمینه ، در اطراف نمادهای کد شده است. به طور معمول ، از زمینه علی (محدود به نمادهایی که قبلاً کدگذاری شده اند) استفاده می شود. هر دو کدگذار (Pennebaker) Q و همکاران [۱۹۸۸] و MQ-coder (ISO/IEC [۲۰۰۰]) ، دو تکنیک کدگذاری حسابی که در JBIG ۲۰۰۰-JPEG و سایر تصاویر مهم گنجانیده شده اند. استانداردهای فشرده سازی ، از مدل های احتمالی استفاده می کنند که هم تطبیقی و هم وابسته به زمینه هستند. کدگذار Q به طور پویا احتمالات نماد را در طول تجدیدنظرهای داخلی که بخشی از فرایند کدگذاری حسابی هستند به روز می کند. مدل های وابسته به متن تطبیقی نیز در کدگذاری Golomb ، به عنوان مثال ، در استاندارد فشرده سازی JPEG-LS استفاده شده است.

شکل ۸.۱۳ (الف) مراحل مربوط به کدگذاری منطقی تطبیقی و وابسته به زمینه نمادهای منبع دودویی را نشان می دهد. کدگذاری حسابی اغلب در هنگام کدگذاری نمادهای دوتایی مورد استفاده قرار می گیرد. هنگامی که هر

نماد (یا بیت) فرآیند کدگذاری را آغاز می کند ، متن آن در بلوک تعیین زمینه شکل ۸.۱۳ (الف) شکل می گیرد. شکل ۸.۱۳ (ب) تا (د) سه زمینه احتمالی را می توان مورد استفاده قرار داد: (۱) نماد بلافاصله قبل ، (۲) گروهی از نمادهای قبلی ، و (۳) تعدادی از نمادهای قبلی به علاوه نمادهایی در خط اسکن قبلی برای سه مورد نشان داده شده ، بلوک تخمین احتمال باید ۲۱ (یا ۲) ، ۲۸ (یا ۲۵۶) و ۲۵ (یا ۳۲) زمینه و احتمالات مربوط به آنها را مدیریت کند. به عنوان مثال ، اگر در شکل ۸.۱۳ (b) استفاده شود ، احتمالات شرطی  $P(0|a=0)$  (احتمال اینکه نماد کد شده ۰ باشد با توجه به اینکه نماد قبلی ۰ است) ،  $P(1|a=1)$  ،  $P(0|a=1)$  ، and  $P(1|a=0)$  یک ۱ - باید پیگیری شود احتمالات مناسب بعداً به عنوان تابعی از زمینه فعلی به بلوک رمزگذاری حسابی منتقل می شود و تولید توالی خروجی کدگذاری شده حسابی را مطابق با فرآیند نشان داده شده در شکل ۸.۱۲ انجام می دهد. سپس احتمالات مربوط به زمینه مربوط به مرحله کدگذاری فعلی به روز می شود تا این واقعیت را نشان دهد که نماد دیگری در آن زمینه پردازش شده است. در نهایت ، ما توجه داریم که انواع روش های رمزنگاری حسابی توسط اختراعات ایالات متحده محافظت می شود (و ممکن است در حوزه های قضایی دیگر نیز محافظت شود).

زیرا



شکل ۸.۱۳ (الف) رویکرد کدگذاری حسابی مبتنی بر متن (اغلب برای نمادهای منبع دودویی استفاده می شود).  
 (ب) - (د) سه مدل زمینه احتمالی.

از جمله این اختراعات ، و امکان قضاوت های پولی نامطلوب برای نقض آنها ، اکثر پیاده سازی های استاندارد فشرده سازی JPEG ، که شامل گزینه هایی برای کدگذاری هافمن و حساب است ، معمولاً از کدگذاری هافمن به تنهایی پشتیبانی می کند.

## ۸.۵ کدگذاری LZW

روشهای پوشش داده شده در بخشهای ۸.۲ تا ۸.۴ بر حذف افزونگی برنامه نویسی متمرکز شده است. در این بخش ، ما یک روش فشرده سازی بدون خطا را در نظر می گیریم که افزونگی های فضایی در یک تصویر را نیز مورد بررسی قرار می دهد. این تکنیک که کدگذاری (LZW) Lempel-Ziv-Welch نامیده می شود ، کلمات کد طول ثابت را به دنباله های طول متغیر نمادهای منبع اختصاص می دهد. از بخش قبلی در مورد اندازه گیری اطلاعات تصویر یادآوری کنید که شانون از ایده کدگذاری توالی نمادهای منبع به جای نمادهای منبع جداگانه در اثبات قضیه اول خود استفاده کرد. یکی از ویژگیهای کلیدی برنامه نویسی LZW این است که به دانش پیشینی در مورد احتمال وقوع نمادهایی که باید کد شوند نیاز ندارد. علیرغم این واقعیت که تا همین اواخر تحت ثبت اختراع ایالات متحده محافظت می شد ، فشرده سازی LZW در انواع قالب های اصلی تصویربرداری از جمله GIF ، TIFF و PDF ادغام شده است. قالب PNG برای برآوردن الزامات مجوز LZW ایجاد شده است.

با ارجاع به جداول ۸.۳-۸.۵ ، کدگذاری LZW در آن استفاده می شود • GIF • TIFF • PDF در هیچ یک از استانداردهای فشرده سازی مجاز بین المللی

مثال ۸.۶: کدگذاری LZW شکل ۸.۹ (الف).

دوباره تصویر ۸ بیتی  $512 * 512$  از شکل ۸.۹ (الف) را در نظر بگیرید. با استفاده از Adobe Photoshop ، نسخه فشرده نشده TIFF این تصویر به  $286,740$  بایت فضای دیسک نیاز دارد- $262,144$  بایت برای  $512 * 512$  پیکسل ۸ بیتی به اضافه  $24,596$  بایت سربار. با استفاده از گزینه فشرده سازی LZW TIFF ، فایل حاصل  $224,420$  بایت است. نسبت تراکم  $CC = 1.28$  است. به یاد بیایید که برای هافمن نشان داده شده در شکل ۸.۹ (a) در مثال ۸.۴ ،  $C = 1.077$  ، کدگذاری شده است. فشرده سازی اضافی با روش LZW به دلیل حذف برخی از افزونگی فضایی تصویر است. کدگذاری LZW از نظر مفهومی بسیار ساده است (ولش [۱۹۸۴]). در آغاز فرایند کد نویسی ، یک کتاب کد یا فرهنگ لغت حاوی نمادهای منبع برای کدگذاری ساخته می شود. برای تصاویر تک رنگ ۸ بیتی ، اولین  $256$  کلمه فرهنگ لغت به شدت  $0$  ،  $1$  ،  $2$  ، ... ،  $255$  اختصاص داده می شود. همانطور که رمزگذار پیکسل های تصویر را به طور پی در پی بررسی می کند ، توالی های شدیدی که در فرهنگ لغت وجود ندارد در الگوریتم قرار می گیرد. مکانهای تعیین شده (به عنوان مثال ، بلااستفاده بعدی). اگر دو پیکسل اول تصویر سفید باشد ، به عنوان مثال ، دنباله "۲۵۵-۲۵۵" ممکن است به مکان  $256$  اختصاص داده شود ، آدرس زیر مکانهایی است که برای سطوح شدت  $0$  تا  $255$  محفوظ است. دفعه بعد که دو پیکسل پیاپی سفید با یکدیگر مواجه می شوند ، کلمه کد  $256$  ، آدرس محل حاوی دنباله  $255-255$  ، برای نشان دادن آنها استفاده می شود. اگر از یک فرهنگ لغت ۹ بیتی و  $512$  کلمه ای در فرایند کدگذاری استفاده شود ، بیت های اصلی (۸ ۸ ۸) که برای نشان دادن دو پیکسل استفاده می شد با یک کلمه کد ۹ بیتی جایگزین می شود. بدیهی

است که اندازه فرهنگ لغت یک پارامتر مهم سیستم است. اگر بسیار کوچک باشد، تشخیص دنباله های سطح شدت تطبیق پذیر کمتر خواهد بود. اگر بزرگ باشد، اندازه کلمات کد بر تابع فشرده سازی تأثیر منفی می گذارد.

### مثال ۸.۷: کدگذاری LZW

تصویر زیر ۴ \* ۴ ، ۸ بیتی از لبه عمودی را در نظر بگیرید:

```
39 39 126 126
39 39 126 126
39 39 126 126
39 39 126 126
```

جدول ۸.۸ مراحل مربوط به کدگذاری ۱۶ پیکسل آن را توضیح می دهد. یک فرهنگ لغت ۵۱۲ کلمه ای با محتوای شروع زیر فرض می شود:

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

مکانهای ۲۵۶ تا ۵۱۱ در ابتدا بلااستفاده هستند.

تصویر با پردازش پیکسل های آن به صورت چپ به راست، از بالا به پایین کدگذاری می شود. هر مقدار شدت متوالی با یک متغیر، ستون ۱ جدول ۸.۸، که "دنباله فعلی شناخته شده" نامیده می شود، ترکیب شده است. همانطور که مشاهده می شود، این متغیر در ابتدا تهی است.

فرهنگ لغت برای هر دنباله متصل جستجو می شود و در صورت یافتن، همانطور که در ردیف اول جدول مشاهده شد، با دنباله ای که اخیراً به هم پیوسته و شناخته شده است (یعنی در فرهنگ لغت) جایگزین می شود. این کار در ستون ۱ ردیف ۲ انجام شد. هیچ کد خروجی تولید نمی شود و فرهنگ لغت تبدیل نمی کند. اگر دنباله متصل شده یافت نشود، آدرس دنباله فعلی شناخته شده به عنوان مقدار رمزگذاری شده بعدی خروجی داده می شود، دنباله متصل اما ناشناخته به فرهنگ لغت اضافه می شود و دنباله فعلی به مقدار پیکسل فعلی مقداردهی می شود. به این در ردیف ۲ جدول رخ داد. دو ستون آخر توالی های شدت را که هنگام اسکن کل تصویر ۱۲۸ بیتی به فرهنگ لغت اضافه می شود، توضیح می دهد. نه کلمه رمز اضافی تعریف شده است. در پایان کدگذاری، فرهنگ لغت شامل ۲۶۵ کلمه کد است و الگوریتم LZW با موفقیت چندین دنباله شدت تکراری را شناسایی کرده است - با استفاده از آنها تصویر ۱۲۸ بیتی اصلی را به ۹۰ بیت (یعنی ۱۰ کد ۹ بیتی) کاهش می دهد. خروجی رمزگذاری شده با خواندن ستون سوم از بالا به پایین بدست می آید. نسبت فشرده سازی حاصل ۱.۴۲:۱ است.

یک ویژگی منحصر به فرد کدگذاری LZW که نشان داده شد این است که دیکشنری کد یا کتاب کد در حالی که داده ها در حال کدگذاری هستند ایجاد می شود. به طور قابل توجهی، یک رمزگشای LZW یک دیکشنری

رفع فشار را یکسان می سازد ، زیرا به طور همزمان جریان داده های رمزگذاری شده را رمزگشایی می کند. رمزگشایی خروجی مثال قبلی و بازسازی کتاب کد به عنوان یک تمرین برای خواننده (به مشکل ۸.۲۰ مراجعه کنید) واگذار می شود. اگرچه در این مثال نیازی نیست ، اما اکثر برنامه های کاربردی به یک استراتژی برای مدیریت بیش از پیش فرهنگ لغت نیاز دارند. یک راه حل ساده این است که وقتی کامل می شود فرهنگ لغت را فعال یا دوباره راه اندازی کنید و با یک دیکشنری جدید ، برنامه نویسی را ادامه دهید. یک گزینه مختلط تر این است که تابع فشرده سازی را تحت نظر داشته باشید و در صورت ضعیف یا غیرقابل قبول بودن فرهنگ لغت را فعال کنید. از طرف دیگر ، می توان کمترین مدخل فرهنگ لغت را ردیابی و در صورت لزوم جایگزین کرد.

جدول ۸.۸ مثال برنامه نویسی LZW.

ورود فرهنگ لغت	محل فرهنگ لغت (کد واژه)	خروجی رمزگذاری	پیکسل در حال پردازش است	دنباله فعلی به رسمیت شناخته شده
			39	
39-39	256	39	39	39
39-126	257	39	126	39
126-126	258	126	126	126
126-39	259	126	39	126
			39	39
39-39-126	260	256	126	39-39
			126	126
126-126-39	261	258	39	126-126
			39	39
39-39-126-126	262	260	126	39-39
			39	126
126-39-39	263	259	39	126-39
			126	39
39-126-126	264	257	126	39-126
			126	126

## ۸.۶ کدگذاری در طول اجرا

همانطور که قبلاً ذکر شد ، تصاویر با شدتهای تکراری در طول ردیفها (یا ستونها) اغلب می توانند با نمایش شدتهای یکسان به عنوان جفتهای طول اجرا فشرده شوند ، جایی که هر جفت طول اجرا شروع شدت جدید و تعداد را مشخص می کند. پیکسل های متوالی که دارای این شدت هستند. این تکنیک ، که به آن کدگذاری طول اجرا (RLE) گفته می شود ، در دهه ۱۹۵۰ توسعه یافت و همراه با پسوندهای دو بعدی آن ، به روش استاندارد فشرده سازی در برنامه نویسی فکس (FAX) تبدیل شد. فشار با حذف یک شکل ساده از افزونگی فضایی - گروه

هایی با شدت های یکسان - به دست می آید. وقتی تعداد پیکسل های یکسان (یا تعداد کمی) وجود داشته باشد ، رمزگذاری طول اجرا منجر به گسترش داده ها می شود.

با ارجاع به جداول ۸.۳-۸.۵ ، کدگذاری طول اجرا در استفاده می شود. • CCITT • JBIG۲ • JPEG • M-JPEG • BMP • MPEG-۱،۲،۴ و سایر استانداردها و قالب های فایل فشرده سازی.

#### مثال ۸.۸: RLE در قالب فایل BMP.

قالب BMP از شکلی از کدگذاری طول اجرا استفاده می کند که در آن داده های تصویر در دو حالت مختلف نمایش داده می شود: کدگذاری شده و مطلق. هر دو حالت می توانند در هر نقطه از تصویر رخ دهند. در حالت رمزگذاری شده ، از نمای RLE دو بایت استفاده می شود. اولین بایت تعداد پیکسل های متوالی که دارای شاخص رنگ در بایت دوم هستند را مشخص می کند. شاخص رنگ ۸ بیتی شدت اجرا (رنگ یا مقدار خاکستری) را از جدولی از ۲۵۶ شدت ممکن انتخاب می کند. در حالت مطلق ، اولین بایت ۰ است و بایت دوم یکی از چهار حالت ممکن را نشان می دهد ، همانطور که در جدول ۸.۹ نشان داده شده است. وقتی بایت دوم ۰ یا ۱ باشد ، انتهای یک خط یا انتهای تصویر رسیده است. اگر ۲ باشد ، دو بایت بعدی حاوی افست افقی و عمودی بدون علامت در موقعیت مکانی (و پیکسل) جدید در تصویر است. اگر بایت دوم بین ۳ تا ۲۵۵ باشد ، تعداد پیکسل های فشرده نشده ای را مشخص می کند که با هر بایت بعدی دارای شاخص رنگ یک پیکسل است. تعداد کل بایت ها باید در یک کلمه ۱۶ بیتی تراز شود.

یک فایل BMP فشرده (ذخیره شده با استفاده از فتوشاپ) از تصویر ۵۱۲\*۸۵۱۲ بیتی که در شکل ۸.۹ (a) نشان داده شده است ، به ۲۶۳،۲۴۴ بایت حافظه نیاز دارد. فشرده سازی با استفاده از گزینه RMP BMP ، فایل به ۲۶۷،۷۰۶ بایت گسترش می یابد و نسبت فشرده سازی  $C = 0.98$  است. دوامهای با شدت مساوی کافی برای موثر بودن فشرده سازی طول اجرا وجود ندارد. مقدار کمی از انبساط رخ می دهد. برای تصویر در شکل ۸.۱ (C) ، با این حال ، گزینه BMP RLE منجر به نسبت فشرده سازی  $C = 1.35$  می شود. (توجه داشته باشید که به دلیل تفاوت در سربار ، فایل BMP فشرده نشده کوچکتر از فایل TIFF فشرده نشده در مثال ۸.۶ است.) رمزگذاری طول اجرا به ویژه هنگام فشرده سازی تصاویر باینری موثر است. از آنجا که تنها دو شدت ممکن (سیاه و سفید) وجود دارد ، پیکسل های مجاور به احتمال زیاد یکسان هستند. علاوه بر این ، هر ردیف تصویر را می توان تنها با دنباله ای از طول ها نشان داد ، نه جفت های شدت طول ، مانند مثال ۸.۸. ایده اصلی این است که هر گروه پیوسته (یعنی اجرا) از ۰ یا ۱ را که در یک اسکن چپ به راست یک ردیف با طول آن قرار گرفته است کدگذاری کنید و قراردادی برای تعیین مقدار اجرا ایجاد کنید. رایج ترین قراردادهای عبارتند از (۱) تعیین مقدار اولین اجرای هر ردیف ، یا (۲) فرض بر این است که هر سطر با یک دور سفید شروع می شود ، که طول اجرا در واقع ممکن است صفر باشد. اگرچه رمزگذاری طول اجرا به خودی خود یک روش موفثر برای فشرده سازی تصاویر دوتایی است ، اما می توان با کدگذاری طول متغیر خود طول اجرا ، فشرده سازی بیشتری را به دست آورد. طول های



سیاه و سفید را می توان به طور جداگانه با استفاده از کدهای طول متغیر که به طور خاص مطابق با آمار خودشان کدگذاری شده اند کدگذاری کرد. به عنوان مثال ، اگر اجازه دهیم  $a_j$  نشان دهنده یک سیاه و سفید با طول  $j$  باشد ، ما می توانیم این احتمال را که نماد  $a_j$  توسط یک منبع سیاه و سفید طول سیاه منتشر شده است را با تقسیم تعداد طول سیاه طول  $j$  در کل تصویر بر کل تخمین بزنیم.

تعداد دویدن سیاه برآوردی از آنتروپی این منبع سیاه و سفید طول ، نشان داده شده به عنوان  $H_0$  ، با جایگزینی این احتمالات به معادله (۸-۶) می آید. استدلال مشابهی در مورد آنتروپی دوره های سفید وجود دارد که به عنوان  $H_1$  نشان داده می شود. آنتروپی تقریبی طول اجرا تصویر است

$$H_{RL} = \frac{H_0 + H_1}{L_0 + L_1} \quad (8-18)$$

جایی که متغیرهای  $L_0$  و  $L_1$  به ترتیب مقدار متوسط طول سیاه و سفید را نشان می دهند. معادله (۸-۱۸) برآوردی از میانگین تعداد بیت در پیکسل مورد نیاز برای کدگذاری طول اجرا در یک تصویر باینری با استفاده از یک کد با طول متغیر ارائه می دهد.

جدول ۸.۹ گزینه های حالت برنامه نویسی مطلق BMP. در این حالت ، اولین بایت جفت  $BMP_0$  است.

مقدار بیت دوم	e	وضعیت
0		انتهای خط
1		انتهای تصویر
2		حرکت به موقعیت جدید
3-255		پیکسل ها را به صورت جداگانه

دو مورد از قدیمی ترین و پرکاربردترین استانداردهای فشرده سازی تصویر ، استانداردهای گروه ۳ و ۴ CCITT برای فشرده سازی تصویر باینری است. اگرچه در انواع برنامه های رایانه ای مورد استفاده قرار گرفته اند ، اما در اصل به عنوان روش های کدگذاری فاکس (FAX) برای تبدیل اسناد از طریق شبکه های تلفن طراحی شده اند. استاندارد گروه ۳ از تکنیک کدگذاری  $D-1$  طول اجرا استفاده می کند که در آن آخرین خطوط  $K-1$  از هر گروه از خطوط  $K$  (برای  $K = 2$  یا  $4$ ) می تواند به صورت اختیاری به صورت دو بعدی کدگذاری شود. استاندارد گروه ۴ یک نسخه ساده یا ساده از استاندارد گروه ۳ است که در آن فقط کدگذاری دو بعدی مجاز است. هر دو استاندارد از روش کدگذاری دو بعدی استفاده می کنند ، که دو بعدی است به این معنا که اطلاعات خط قبلی برای کدگذاری خط فعلی استفاده می شود. کدگذاری  $D-1$  و  $D-2$  در ادامه مورد بحث قرار می گیرد.

### فشرده سازی یک بعدی CCITT

در استاندارد فشرده سازی ۳-۱ C CCITT Group ، هر خط از تصویر<sup>۳۶</sup> به صورت مجموعه ای از کلمات رمز Huffman با طول متغیر کدگذاری شده است که طول اجرای متغیرهای اجرای سفید و سیاه را در اسکن چپ به راست خط روش فشرده سازی مورد استفاده معمولاً کدگذاری Modified Huffman (MH) است. خود کلمات رمز بر دو نوع هستند که استاندارد آنها را به عنوان کدهای پایان دهنده و کدهای آرایش نام می برد. اگر طول اجرا ۲ کمتر یا مساوی ۶۳ باشد ، از کد پایانی برای نمایش آن استفاده می شود. این استاندارد کدهای مختلف خاتمه برای رنگ مشکی را مشخص می کند و دویدن سفید اگر  $r = 63$  ، از دو کد استفاده می شود. یک کد آرایش برای ضریب  $\lfloor r/64 \rfloor \times 64$  و یک کد خاتمه برای بقیه  $r \bmod 64$ . کدهای آرایش ممکن است بستگی به شدت (سیاه یا سفید) اجرای کدگذاری شده داشته باشد. اگر  $\lfloor r/64 \rfloor \times 64 \leq 1728$  ، کدهای آرایش سیاه و سفید جداگانه مشخص شده است. در غیر این صورت ، کدهای آرایش هستند

مستقل از شدت دویدن این استاندارد مستلزم این است که هر خط با یک کلمه کد سفید طول اجرا شروع شود ، که در واقع ممکن است ۰۰۱۱۰۱۰۱ باشد ، کد یک دور سفید با طول صفر. در نهایت ، یک کلمه کد منحصر به فرد پایان خط ۰۰۰۰۰۰۰۰۰۰۱ (EOL) برای خاتمه هر خط و همچنین برچسب گذاری خط اول هر تصویر جدید استفاده می شود. انتهای دنباله ای از تصاویر با شش EOL متوالی نشان داده می شود.

برای مشاهده جداول کدهای پایان دهنده و آرایش MH به وب سایت کتاب مراجعه کنید. به یاد داشته باشید که نماد [X] بزرگترین میانجی کوچکتر یا مساوی X را نشان می دهد

### فشرده سازی CCITT دو بعدی

رویکرد فشرده سازی دو بعدی برای استانداردهای گروه ۳ و ۴ CCITT اتخاذ شده است ، یک روش خط به خط است که در آن موقعیت هر یک از تبدیل سیاه به سفید یا سفید به سیاه به نسبت کدگذاری می شود. موقعیت عنصر مرجع  $a_0$  که در خط کدگذاری فعلی قرار دارد. خط قبلی کدگذاری شده خط مرجع نامیده می شود. خط مرجع خط اول هر تصویر جدید یک خط سفید موهومی است. تکنیک کدگذاری دو بعدی که استفاده می شود ، کدگذاری Relative عنصر (READ) Designate نامیده می شود. در استاندارد گروه ۳ ، یک یا سه خط READ کدگذاری شده بین خطوط پی در پی کد MH مجاز است. این تکنیک کدگذاری اصلاح شده READ (MR) نامیده می شود. در استاندارد گروه ۴ ، تعداد بیشتری از خطوط کدگذاری READ مجاز است و این روش کدگذاری Modified Modified READ (MMR) نامیده می شود. همانطور که قبلاً اشاره شد ، کدگذاری دو بعدی است به این معنا که اطلاعات خط قبلی برای کدگذاری خط فعلی استفاده می شود. دگرگونی های دو بعدی دخیل نیستند. خط سفید موهومی تکنیک کدگذاری دو بعدی که مورد استفاده قرار می گیرد ، کدگذاری Relative عنصر (READ) Designate نامیده می شود. در استاندارد گروه ۳ ، یک یا سه خط READ کدگذاری

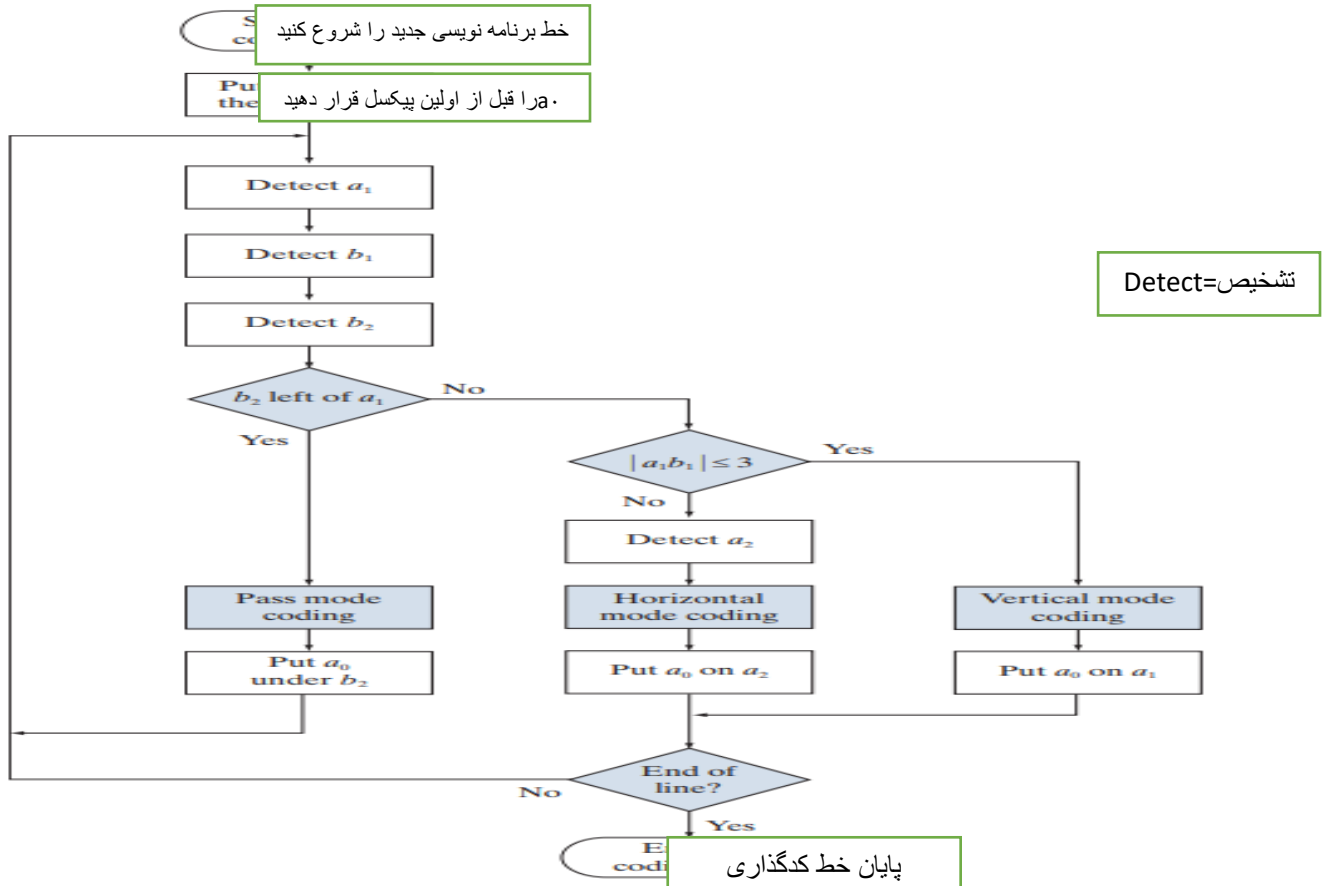
<sup>۳۶</sup> در استاندارد ، تصاویر را صفحات و دنباله های تصاویر را اسناد می نامند.

شده بین خطوط پی در پی کد MH مجاز است. این تکنیک کدگذاری اصلاح شده READ (MR) نامیده می شود. در استاندارد گروه ۴، تعداد بیشتری از خطوط کدگذاری READ مجاز است و این روش کدگذاری Modified Modified READ (MMR) نامیده می شود. همانطور که قبلاً ذکر شد، کدگذاری دو بعدی است به این معنا که اطلاعات خط قبلی برای کدگذاری خط فعلی استفاده می شود. دگرگونی های دو بعدی دخیل نیستند. شکل ۸.۱۴ فرایند اصلی کدگذاری دو بعدی را برای یک خط اسکن واحد نشان می دهد. توجه داشته باشید که مراحل اولیه روش برای تعیین چند عنصر تبدیل دهنده کلیدی است:  $a_0$ ،  $a_1$ ،  $a_2$ ،  $b_1$  و  $b_2$ . یک عنصر در حال تبدیل توسط استاندارد به عنوان پیکسلی تعریف می شود که مقدار آن متفاوت از پیکسل قبلی در یک خط است. مهمترین عنصر در حال تبدیل  $a_0$  (عنصر مرجع) است که یا در محل یک عنصر تبدیل رنگ سفید در سمت چپ اولین پیکسل هر خط کدگذاری جدید تنظیم می شود یا از حالت کدگذاری قبلی تعیین می شود. حالت های کدگذاری در پاراگراف زیر مورد بحث قرار می گیرد. پس از قرار گرفتن  $a_0$ ،  $a_1$  به عنوان محل تبدیل عنصر بعدی در سمت راست  $a_0$  در خط کدگذاری فعلی،  $a_2$  به عنوان عنصر تبدیل بعدی در سمت راست  $a_1$  در خط کدگذاری،  $b_1$  به عنوان عنصر تبدیل دهنده مشخص می شود. با مقدار مخالف ( $a_0$ ) و سمت راست  $a_0$  در خط مرجع (یا قبلی)، و  $b_2$  به عنوان عنصر تبدیل بعدی در سمت راست  $b_1$  در خط مرجع. اگر هر یک از این عناصر در حال تبدیل تشخیص داده نشوند، در محل پیکسل موهومی در سمت راست آخرین پیکسل در خط مناسب تنظیم می شوند. شکل ۸.۱۵ دو تصویر از روابط عمومی بین عناصر مختلف در حال تبدیل ارائه می دهد. پس از شناسایی عنصر مرجع فعلی و تبدیل عناصر مرتبط، دو آزمایش ساده برای انتخاب یکی از سه حالت کدگذاری احتمالی انجام می شود: حالت گذر، حالت عمودی یا حالت افقی. آزمایش اولیه، که مربوط به اولین نقطه شاخه در owchart در شکل ۸.۱۴ است، محل  $b_2$  را با  $a_1$  مقایسه می کند. دومین آزمایش، که مربوط به نقطه شاخه دوم در شکل ۸.۱۴ است، فاصله (بر حسب پیکسل) بین مکان های  $a_1$  و  $b_1$  را محاسبه کرده و آن را با ۳ مقایسه می کند. بسته به نتیجه این آزمایشها، یکی از سه کدگذاری مشخص شده بلوک های شکل ۸.۱۴ وارد شده و روش کدگذاری مناسب اجرا می شود. سپس بر اساس نمودار خروجی، یک عنصر مرجع جدید در آماده سازی برای تکرار کدگذاری بعدی ایجاد می شود. جدول ۸.۱۰ کدهای خاصی را که برای هر یک از سه حالت کدگذاری احتمالی استفاده شده است، مشخص می کند.

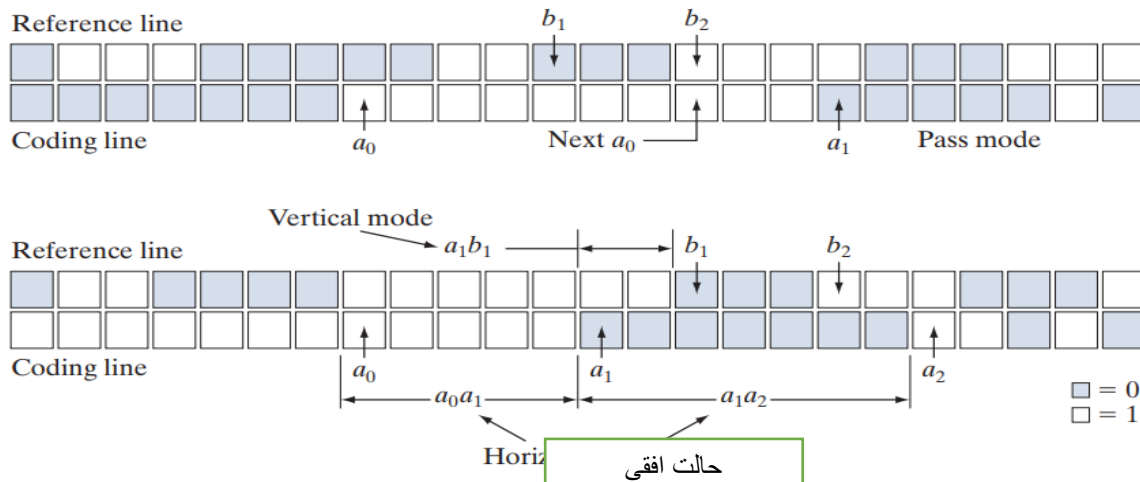
در حالت گذر، که به طور خاص موردی را که در آن  $b_2$  مستقیماً بالای  $a_1$  قرار دارد مستثنی می کند، تنها کلمه کد گذر حالت ۰۰۰۱ مورد نیاز است. همانطور که در شکل ۸.۱۵ (الف) نشان داده می شود، این حالت خطوط مرجع سفید یا سیاه را مشخص می کند که با خطوط کدگذاری سفید یا سیاه فعلی همپوشانی ندارند. در حالت کدگذاری افقی، فاصله های  $a_0$  تا  $a_1$  و  $a_1$  تا  $a_2$  باید مطابق با کدهای خاتمه و آرایش فشرده سازی ۱-۳ D CCITT Group کدگذاری شود، سپس به کلمه کد حالت افقی ۰۰۱ اضافه شود. این در جدول نشان داده شده است. ۸.۱۰ با نماد  $001 + M(a_1a_2) + M(a_0a_1)$ ، که در آن  $a_0a_1$  و  $a_1a_2$  به ترتیب فاصله های

$a_0$  تا  $a_1$  و  $a_1$  تا  $a_2$  را نشان می دهند. سرانجام ، در حالت کدگذاری عمودی ، یکی از شش کد طول متغیر ویژه به فاصله بین  $a_1$  و  $b_1$  اختصاص داده می شود. شکل ۸.۱۵ (ب) پارامترهای دخیل در کدگذاری حالت افقی و عمودی را نشان می دهد. کلمه کد حالت افزودنی در پایین جدول ۸.۱۰ برای ورود به حالت کدگذاری فکس اختیاری استفاده می شود. برای مثال ، کد ۰۰۰۰۰۰۱۱۱۱ برای شروع حالت تبدیل فشرده نشده استفاده می شود.

برای اطلاع از جداول کدگذاری استاندارد CCITT به وب سایت کتاب مراجعه کنید



شکل ۸.۱۴ ۲-D CCITT روش کد نویسی را بخوانید. علامت  $a_1b_1$  مقدار مطلق فاصله بین عناصر در حال تبدیل  $a_1$  را نشان می دهد و  $b_1$



شکل ۸.۱۵ (a) CCITT حالت عبور و (ب) پارامترهای کدگذاری حالت افقی و عمودی.

مثال ۸.۹: مثال کدگذاری حالت عمودی CCITT.

اگرچه شکل ۸.۱۵ (ب) با پارامترهای کدگذاری حالت افقی و عمودی (برای تسهیل بحث در بالا) ذکر شده است ، الگوی نشان داده شده از پیکسل های سیاه و سفید موردی برای کدگذاری حالت عمودی است. بدین معنا که چون  $b_2$  در سمت راست  $a_1$  است ، اولین آزمایش (یا حالت قبولی) در شکل ۸.۱۴ شکست می خورد. آزمون دوم که تعیین می کند حالت کدگذاری عمودی یا افقی وارد شده است یا خیر ، نشان می دهد که باید از کدگذاری حالت عمودی استفاده شود ، زیرا فاصله  $a_1$  تا  $b_1$  کمتر از ۳ است. مطابق جدول ۸.۱۰ ، کلمه کد مناسب ۰۰۰۰۱۰ است ، به این معنی که  $a_1$  دو پیکسل از  $b_1$  باقی مانده است. در آماده سازی برای تکرار کدگذاری بعدی ،  $a_0$  به محل  $a_1$  منتقل می شود. جدول ۸.۱۰ CCITT جدول کد دو بعدی

Mode	Code Word
عبور	0001
افقی	$001 + M(a_0a_1) + M(a_1a_2)$
عمودی	
زیر $b_1$	1
$a_0$ یکی به راست of $b_1$	011
$a_0$ دو تا به راست of $b_1$	000011
$a_0$ سه تا به راست of $b_1$	0000011
$a_0$ یکی به چپ of $b_1$	010
$a_0$ دو تا به چپ of $b_1$	000010
$a_0$ سه تا به چپ of $b_1$	0000010
افزونه n	0000001xxx

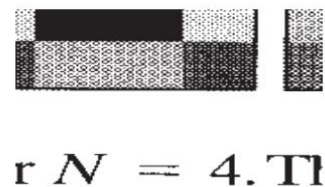
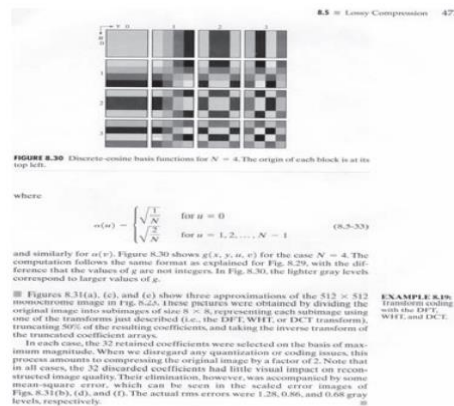
## مثال ۸.۱۰: مثال فشرده سازی CCITT.

شکل ۸.۱۶ (a) یک اسکن ۳۰۰ dpi از صفحه کتاب ۷ \* ۹.۲۵ اینچی است که در مقیاس حدود ۳۱ نمایش داده می شود. توجه داشته باشید که تقریباً نیمی از صفحه حاوی متن است، حدود ۹ توسط یک تصویر نیم تنه اشغال شده است و بقیه فضای سفید است. بخشی از صفحه در شکل ۸.۱۶ (ب) بزرگ شده است. به خاطر داشته باشید که ما با یک تصویر باینری سروکار داریم. توهم تن های خاکستری، همانطور که در بخش ۴.۵ توضیح داده شد، توسط فرآیند نیم تنه مورد استفاده در چاپ ایجاد می شود. اگر پیکسل های دوتایی تصویر در شکل ۸.۱۶ (a) در گروه های ۸ پیکسل در هر بایت ذخیره شوند، تصویر اسکن شده ۲۶۹۷ \* ۱۹۵۲، که معمولاً سند نامیده می شود، به ۶۵۸،۰۶۸ بایت نیاز دارد. یک فایل PDF فشرده نشده از سند (ایجاد شده در Photoshop) به ۶۶۳،۴۴۵ بایت نیاز دارد. فشرده سازی گروه ۳ CCITT، فایل را به ۱۲۳،۴۹۷ بایت کاهش می دهد و در نتیجه نسبت فشرده سازی  $C = 5,37$  می شود. فشرده سازی گروه ۴ CCITT، فایل را به ۱۱۰،۴۵۶ بایت کاهش می دهد و نسبت فشرده سازی را به حدود ۶ افزایش می دهد.

## ۸.۷ کدگذاری مبتنی بر سیمبول

در کدگذاری مبتنی بر نمادها یا نشانه ها، یک تصویر به عنوان مجموعه ای از تصاویر فرعی متداول که نمادها نامیده می شوند، نشان داده می شود. هر چنین نمادی در یک نماد ذخیره می شود فرهنگ لغت و تصویر بصورت مجموعه ای از سه قلو  $\{(x_1, y_1, t_1), (x_2, y_2, t_2), \dots\}$ ، جایی که هر جفت  $(y_i, x_i)$  محل یک نماد را در تصویر مشخص می کند و نشانه  $t$  آدرس آدرس نماد یا زیر تصویر در فرهنگ لغت است. یعنی هر سه گانه نمونه ای از نماد فرهنگ لغت را در تصویر نشان می دهد. ذخیره یکبار همگن فقط یکبار می تواند تصاویر را بطور قابل توجهی فشرده کند، به ویژه در ذخیره اسناد و برنامه های بازیابی که در آن نمادها اغلب bitmap های شخصیت هستند که بارها تکرار می شوند.

با توجه به جداول ۸.۳-۸.۵، از کدگذاری مبتنی بر نماد در استفاده می شود • فشرده سازی JBIG۲.



شکل ۸.۱۶ اسکن باینری صفحه کتاب: (الف) برای نشان دادن محتوای کلی صفحه مقیاس بندی شده است. (ب) مقیاس بندی شده است تا پیکسل های دودویی مورد استفاده در dithering را نشان دهد.

تصویر ساده دو سطحی را در شکل ۸.۱۷ (a) در نظر بگیرید. این شامل تنها کلمه ، موز است که از سه نماد منحصر به فرد تشکیل شده است:  $a$  ،  $b$  و  $n$ . با فرض اینکه  $b$  اولین نماد شناسایی شده در فرایند کدگذاری است ، نقشه  $۹ * ۷$  بیتی آن در محل  $۰$  فرهنگ لغت نمادها ذخیره می شود. همانطور که در شکل ۸.۱۷ (b) نشان داده می شود ، نشانه شناسایی نقشه بیت  $۰$   $b$  است.

بنابراین ، اولین سه گانه در نمای تصویر کدگذاری شده [شکل ۸.۱۷ (c)] را ببینید (۰ ، ۲ ، ۰) است ، که نشان می دهد گوشه بالا سمت چپ (یک قرارداد دلخواه) نقشه بیت مستطیلی که نماد  $b$  را نشان می دهد ، باید در محل (۰ ، ۲) تصویر رمزگشایی شده قرار گیرد. پس از مشخص شدن نقشه های بیتی برای نمادهای  $a$  و  $n$  به فرهنگ لغت ، بقیه تصویر را می توان با پنج سه گانه اضافی کدگذاری کرد. تا زمانی که شش سه گانه مورد نیاز برای قرار دادن نمادها در تصویر ، همراه با سه نقشه بیت مورد نیاز برای تعریف آنها ، کوچکتر از تصویر اصلی باشد ، فشرده سازی رخ می دهد. در این حالت ، تصویر شروع دارای  $۹ * ۵۱ * ۱$  یا  $۴۵۹$  بیت است و با فرض اینکه هر سه گانه از سه بایت ، نمای فشرده دارای  $(۶ * ۳ * ۸) + (۶ * ۶) * (۷ * ۶) * (۹ * ۷) * ۲۸۵$  بیت ؛ نسبت فشرده سازی حاصل  $C = ۱,۶۱$  . برای رمزگشایی نماینده مبتنی بر نماد در شکل ۸.۱۷ (ج) ، شما به سادگی نقشه های بیتی نمادهای مشخص شده در سه گانه را از فرهنگ لغت نمادها می خوانید و آنها را در مختصات مکانی مشخص شده در هر سه گانه قرار می دهید. فشرده سازی مبتنی بر نماد در اوایل دهه ۱۹۷۰ پیشنهاد شد (Ascher and Nagy [۱۹۷۴]) ، اما اخیراً کاربردی شده است. پیشرفت در الگوریتم های منطبق با نمادها (به فصل ۱۲ مراجعه کنید) و افزایش سرعت پردازش رایانه CPU ، امکان انتخاب نمادهای فرهنگ لغت و همچنین یافتن به موقع آنها در یک تصویر را ممکن ساخته است. و مانند بسیاری از روشهای فشرده سازی دیگر ، رمزگشایی مبتنی بر نماد به طور قابل توجهی سریعتر از رمزگذاری است. در نهایت ، ما توجه داریم که هم بیت مپ های نمادی که در فرهنگ لغت ذخیره شده اند و هم سه گانه ای که برای ارجاع به آنها استفاده می شود ، می توانند برای بهبود تابع فشرده سازی کدگذاری شوند. اگر ، مانند شکل ۸.۱۷ ، فقط مطابقت دقیق نمادها مجاز باشد ، فشرده سازی حاصله بدون ضرر است. اگر تفاوت های کوچک مجاز باشد ، برخی از خطاهای بازسازی وجود خواهد داشت.

## فشرده سازی JBIG۲

JBIG۲ یک استاندارد بین المللی برای فشرده سازی تصویر دو سطحی است. با تقسیم بندی یک تصویر به مناطق همپوشان و/یا غیر همپوشان متن ، نصف و محتوای عمومی ، از تکنیک های فشرده سازی که به طور خاص برای هر نوع محتوا بهینه شده اند استفاده می شود:



Token	Symbol	Triplet
0		(0, 2, 0) (3, 10, 1) (3, 18, 2) (3, 26, 1) (3, 34, 2) (3, 42, 1)
1		
2		

شکل ۸.۱۷ (الف) یک سند دو سطحی ، (ب) فرهنگ لغت نمادها ، و (ج) سفرهای مورد استفاده برای تعیین نمادها در سند.

۱. نواحی متنی از شخصیت هایی تشکیل شده اند که به طور ایده آل برای رویکرد کدگذاری مبتنی بر نماد مناسب هستند. به طور معمول ، هر نماد با یک bitmap کاراکتر مطابقت دارد - یک تصویر فرعی که نشان دهنده یک متن است. به طور معمول فقط یک بیت مپ (یا زیر تصویر) در فرهنگ لغت نمادها برای هر نویسه بزرگ و کوچک فونت مورد استفاده وجود دارد. به عنوان مثال ، یک بیت مپ "a" در فرهنگ لغت ، یک بیت مپ "A" ، یک بیت مپ "b" و غیره وجود خواهد داشت . در فشرده سازی زیان بار JBIG۲ ، که اغلب بصورت بدون ضرر یا بصری بدون تلفات نامیده می شود ، ما از تفاوت بین بیت مپ های فرهنگ لغت (به عنوان مثال ، بیت مپ های کاراکتر مرجع یا الگوهای کاراکتر) و موارد خاصی از شخصیت های مرتبط در تصویر غافل می شویم.

در فشرده سازی بدون اتلاف ، تفاوتها ذخیره می شوند و همراه با سه قلوهایی که هر نویسه را رمزگذاری می کنند (توسط رمزگشایی) برای تولید بیت مپ های حقیقی تصویر استفاده می شود. همه نقشه های بیتی به صورت حسابی یا با استفاده از MMR رمزگذاری می شوند (بخش ۸.۶ را ببینید). سه قلوهایی که برای دسترسی به نوشته های فرهنگ لغت استفاده می شوند یا حسابی هستند و یا هافمن رمزگذاری شده اند.

۲. ناحیه های نیم تنه شبیه مناطق متنی هستند از این جهت که از الگوهایی تشکیل شده اند که در یک شبکه معمولی مرتب شده اند. با این حال ، نمادهایی که در فرهنگ لغت ذخیره می شوند ، نقشه های بیتی شخصیت نیستند ، بلکه الگوهای دوره ای هستند که نشان دهنده شدت (به عنوان مثال ، یک عکس) هستند که برای تولید تصاویر دو سطحی برای چاپ حذف شده اند.

۳. مناطق عمومی شامل اطلاعات غیر متنی ، بدون نیمه ، مانند خط خطی و نویز هستند و با استفاده از کدگذاری حسابی یا MMR فشرده می شوند.

همانطور که در بسیاری از استانداردهای فشرده سازی تصویر صادق است ، JBIG۲ رفتار رمزگشایی را تعریف می کند. این به صراحت یک رمزگذار استاندارد را تعریف نمی کند ، اما انعطاف پذیری کافی برای اجازه طرح های مختلف رمزگذار را دارد. اگرچه طراحی رمزگذار نامشخص باقی مانده است ، اما مهم است زیرا سطح فشرده سازی به دست آمده را تعیین می کند. پس از همه ، رمزگذار باید تصویر را به مناطق تقسیم کند ، متن و نمادهای نیمه



تنظیمی را که در فرهنگ لغت ها ذخیره شده است انتخاب کند و تصمیم بگیرد که این نمادها اساساً با نمونه های احتمالی نمادهای موجود در تصویر یکسان یا متفاوت هستند. رمزگشایی به سادگی از این اطلاعات برای ایجاد تصویر اصلی استفاده می کند.

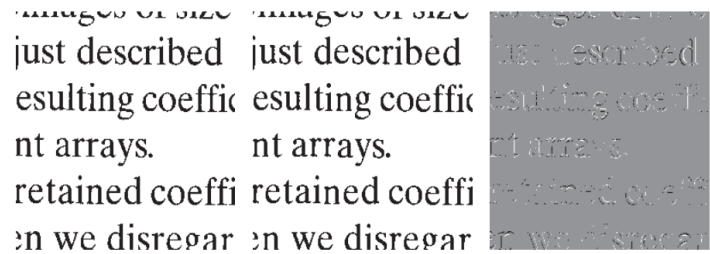
#### مثال ۸.۱۱: مثال فشرده سازی JBIG۲

دوباره تصویر دو سطحی را در شکل ۸.۱۶ (الف) در نظر بگیرید. شکل ۸.۱۸ (الف) یک بخش بازسازی شده از تصویر را پس از کدگذاری JBIG۲ بدون اتلاف (توسط یک برنامه فشرده سازی اسناد تجاری موجود) نشان می دهد. این یک کپی دقیق از تصویر اصلی است. توجه داشته باشید که ds در متن بازسازی شده ، با وجود این که از یک ورودی d در فرهنگ لغت ایجاد شده اند ، کمی متفاوت است. تفاوت بین d و ds در تصویر برای بازنمایی خروجی فرهنگ لغت استفاده شد. استاندارد برای انجام این کار در طول رمزگشایی بیت مپ های فرهنگ لغت رمزگذاری شده ، الگوریتمی را تعیین می کند. به منظور بحث ما ، می توانید آن را به عنوان افزودن تفاوت بین نقشه بیت دیکشنری و نمونه ای خاص از کاراکتر مربوطه در تصویر به نقشه بیت خوانده شده از دیکشنری تصور کنید.

شکل ۸.۱۸ (ب) بازسازی دیگری از ناحیه در شکل ۸.۱۸ (الف) پس از فشرده سازی JBIG۲ بدون ضرر است. توجه داشته باشید که ds در این تصویر یکسان است. آنها مستقیماً از فرهنگ لغت سیمبول کپی شده اند. بازسازی به لحاظ ادراکی بدون ضرر نامیده می شود زیرا متن قابل خواندن است و فونت حتی یکسان است. تفاوت‌های کوچکی که در شکل ۸.۱۸ (ج) بین ds در تصویر اصلی و d در فرهنگ لغت نشان داده شده است ، بی اهمیت تلقی می شوند زیرا بر خوانایی تأثیر نمی گذارد.

به یاد داشته باشید که ما با تصاویر دو سطحی سروکار داریم ، بنابراین در شکل ۸.۱۸ (ج) فقط سه شدت وجود دارد. شدت ۱۲۸ مناطقی را نشان می دهد که هیچ تفاوتی بین پیکسل های مربوط به تصاویر در شکل وجود ندارد. ۸.۱۸ (الف) و (ب) ؛ شدت ۰ (سیاه) و ۲۵۵ (سفید) پیکسل هایی با شدت مخالف در دو تصویر را نشان می دهد - به عنوان مثال ، پیکسل سیاه در یک تصویر که در تصویر دیگر سفید است و برعکس.

فشرده سازی JBIG۲ بدون اتلاف که برای ایجاد شکل ۸.۱۸ (a) استفاده می شود ، ۶۶۳،۴۴۵ بایت اصلی فایل PDF فشرده نشده را به ۳۲،۷۰۵ بایت کاهش می دهد. نسبت تراکم  $C = 20.3$  است. فشرده سازی JBIG۲ بدون ضرر تصویری را به ۲۳،۹۱۳ بایت کاهش می دهد و نسبت فشرده سازی را به حدود ۲۷/۷ افزایش می دهد. این فشرده سازی ۴ تا ۵ برابر بیشتر از CCITT گروه ۳ است و نتایج ۴ از مثال ۸.۱۰.



شکل ۸.۱۸ مقایسه تراکم JBIG۲: (الف) فشرده سازی و بازسازی بدون ضرر. (ب) بدون ادراک بدون ضرر؛ و (ج) تفاوت مقیاس شده بین این دو.

### ۸.۸ کدگذاری BIT-PLANE

تکنیک های اجرا و مبتنی بر نماد بخشهای قبلی را می توان با پردازش جداگانه صفحات بیت خود برای تصاویر با بیش از دو شدت اعمال کرد. این تکنیک که کدگذاری صفحه بیت نامیده می شود، بر اساس مفهوم تجزیه یک تصویر چند سطحی (تک رنگ یا رنگی) به مجموعه ای از تصاویر دوتایی (بخش ۳.۲ را ببینید) و فشرده سازی هر تصویر دودویی از طریق یکی از چندین باینری معروف است. روش های فشرده سازی در این بخش، ما دو روش رایج تجزیه را توصیف می کنیم. شدت یک تصویر تک رنگ  $m$ -bit را می توان در قالب چند جمله ای پایه ۲ نشان داد

با توجه به جداول ۸.۳-۸.۵، از کدگذاری صفحه بیت در استفاده می شود. • JBIG۲ • JPEG-۲۰۰۰ استانداردهای فشرده سازی

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \dots + a_12^1 + a_02^0 \quad (8-19)$$

بر اساس این ویژگی، یک روش ساده برای تجزیه تصویر به مجموعه ای از تصاویر دوتایی این است که ضرایب  $m$  چند جمله ای را به  $m$  سطح ۱ بیتی  $m$  تقسیم کنید. همانطور که در بخش ۳.۲ ذکر شد، صفحه بیت پایین ترین مرتبه (صفحه مربوط به کمترین بیت) با جمع آوری بیت های  $a_0$  هر پیکسل تولید می شود، در حالی که بالاترین مرتبه صفحه بیت شامل بیت ها یا ضرایب  $a_{m-1}$  است. به طور کلی، هر صفحه بیت با تنظیم پیکسل های آن برابر مقادیر بیت های مناسب یا ضرایب چند جمله ای از هر پیکسل در تصویر اصلی ساخته می شود. نقطه ضعف ذاتی این روش تجزیه این است که تبدیلات کوچک در شدت می تواند تأثیر بسزایی بر پیچیدگی صفحات بیت داشته باشد. اگر پیکسل شدت ۱۲۷ (۰۱۱۱۱۱۱۱) در مجاورت پیکسل شدت ۱۲۸ (۱۰۰۰۰۰۰۰) باشد، برای مثال، هر صفحه بیت دارای یک تبدیل ۰ تا ۱ (یا ۱ به ۰) مربوطه خواهد بود. به عنوان مثال، از آنجا که مهمترین بیت های کدهای دوتایی برای ۱۲۷ و ۱۲۸ متفاوت است، بالاترین صفحه بیت دارای پیکسل با ارزش صفر در کنار پیکسل با مقدار ۱ است و یک تبدیل ۰ به ۱ (یا ۱ به ۰) ایجاد می کند. در آن نقطه، یک روش تجزیه جایگزین (که تأثیر تبدیلات شدت کوچک را کاهش می دهد) این است که ابتدا تصویر را با یک کد

خاکستری m-bit نشان دهیم. کد خاکستری  $g_0, g_1, g_2, \dots, g_{m-1}$  که با چند جمله ای در معادله مطابقت دارد. (۸-۱۹) را می توان از آن استفاده کرد

$$g_i = a_i \oplus a_{i+1} \quad 0 \leq i \leq m-2$$

$$g_{m-1} = a_{m-1}$$
(8-20)

در اینجا ، ۸ نشان دهنده عملیات منحصر به فرد OR است. این کد دارای ویژگی منحصر به فردی است که کلمات کد پی در پی تنها در یک موقعیت بیت متفاوت است. بنابراین ، تبدیلات کوچک در شدت کمتر احتمال دارد بر تمام صفحات m bit تأثیر بگذارد. به عنوان مثال ، هنگامی که سطوح شدت ۱۲۷ و ۱۲۸ در مجاورت یکدیگر قرار دارند ، تنها صفحه بیت با بالاترین مرتبه دارای ترانزیشن ۰ تا ۱ است ، زیرا کدهای خاکستری که با ۱۲۷ و ۱۲۸ مطابقت دارند به ترتیب ۰۱۰۰۰۰۰۰ و ۱۱۰۰۰۰۰۰ است.

با توجه به جداول ۸.۳-۸.۵ ، کدگذاری تبدیل بلوک در استفاده می شود • JPEG • M-JPEG • MPEG-۱، ۲، ۴ •  
 • H.۲۶۱ ، H.۲۶۲ ، H.۲۶۳ و H.۲۶۴ • DV • HDV • VC-۱

مثال ۸.۱۲: کدگذاری سطوح بیتی.

شکل ۸.۱۹ و ۸.۲۰ هشت صفحه بیتی دوتایی و کد خاکستری از تصویر تک رنگ ۸ بیتی کودک را در شکل ۸.۱۹ (الف) نشان می دهد. توجه داشته باشید که صفحات بیت مرتبه بالا به مراتب مختلط تر از نمونه های مرتبه پایین خود هستند. به این معنا که آنها دارای مناطق بزرگ یکنواخت با جزئیات ، شلوغی یا تصادفی بسیار کمتری هستند. علاوه بر این ، صفحات بیتی با کد خاکستری پیچیدگی کمتری نسبت به سطوح بیتی دوتایی مربوطه دارند. هر دو مشاهدات در نتایج کدگذاری JBIG۲ جدول ۸.۱۱ نشان داده شده است. به عنوان مثال ، توجه داشته باشید که نتایج  $a_5$  و  $g_5$  به طور قابل توجهی بزرگتر از فشرده سازی  $a_6$  و  $g_6$  هستند و هر دو  $g_5$  و  $g_6$  کوچکتر از نمونه های  $a_5$  و  $a_6$  خود هستند. این روند در همه جدول ادامه دارد ، به استثنای  $a_0$ . کدگذاری خاکستری به طور متوسط مزیت فشرده سازی را در حدود ۱:۰.۰۶ فراهم می کند. در ترکیب با یکدیگر ، فایل های خاکستری رنگ اصلی تصویر تک رنگ را با ۶۷۸ ، ۶۷۶ ، ۴۷۵ ، ۹۶۴ یا ۱:۰.۴۳ فشرده می کنند. غیر خاکستری-فایل های کدگذاری شده تصویر را با ۶۷۸ ، ۶۷۶ ، ۵۰۳ ، ۹۱۶ یا ۱:۰.۳۵ فشرده می کنند. در نهایت ، ما توجه داریم که دو بیت کم اهمیت در شکل ۸.۲۰ دارای ساختار ظاهری کمی هستند. از آنجا که این برای اکثر تصاویر تک رنگ ۸ بیتی معمول است ، برنامه نویسی بیت-صفحه معمولاً به تصاویر ۶ بیت/پیکسل یا کمتر محدود می شود. JBIG۱ ، سلف JBIG۲ ، چنین محدودیتی را اعمال می کند.

### ۸.۹ کدگذاری تبدیل بلاک

در این بخش ، ما یک تکنیک فشرده سازی را در نظر می گیریم که یک تصویر را به بلوک های کوچک غیر همپوشانی با اندازه مساوی تقسیم می کند (به عنوان مثال ۸ \* ۸) و بلوک ها را به طور مستقل با استفاده از تبدیل ۲ بعدی پردازش می کند. در کدگذاری تبدیل بلوک ، از یک تبدیل برگشت پذیر و خطی (مانند تبدیل

فوریه) برای ترسیم هر بلوک یا زیر تصویر به مجموعه ای از ضرایب تبدیل استفاده می شود ، که سپس کوانتیزه و کدگذاری می شوند. برای اکثر تصاویر ، تعداد قابل توجهی از ضرایب دارای مقادیر کمی هستند و می توان آنها را به صورت کوانتومی درشت (یا به طور کامل حذف کرد) با اعوجاج کمی در تصویر. انواع تبدیل ، از جمله تبدیل فوریه گسسته (DFT) فصل ۴ ، می تواند برای تبدیل داده های تصویر استفاده شود.

جدول ۸.۱۱ نتایج کدگذاری بی ضرر JBIG۲ برای سطوح بیتی باینری و خاکستری از شکل ۸.۱۹ (الف). این نتایج شامل سربار نمایش PDF هر صفحه بیت است.

Coefficient $m$	Binary Code (PDF bits)	Gray Code (PDF bits)	Compression Ratio
7	6,999	6,999	1.00
6	12,791	11,024	1.16
5	40,104	36,914	1.09
4	55,911	47,415	1.18
3	78,915	67,787	1.16
2	101,535	92,630	1.10
1	107,909	105,286	1.03
0	99,753	107,909	0.92

در این بخش ، توجه خود را به subimages مربعی (بیشترین استفاده) محدود می کنیم. فرض بر این است که تصویر ورودی ، در صورت لزوم ، پر شده است ، به طوری که  $M$  و  $N$  مضرب  $n$  هستند.

شکل ۸.۲۱ یک سیستم کدگذاری معمولی تبدیل بلوکی را نشان می دهد. رمزگشا دنباله ای معکوس از مراحل (به استثنای تابع کمی سازی) رمزگذار را اجرا می کند ، که چهار عملیات نسبتاً ساده را انجام می دهد: تجزیه زیر تصویر ، تبدیل ، کمی سازی و کدگذاری. یک تصویر ورودی  $M * N$  ابتدا به تصاویر فرعی با اندازه  $n * n$  تقسیم می شود ، که سپس برای ایجاد آرایه های تبدیل زیر تصویر  $MN n^2$  ، هر یک به اندازه  $n * n$  ، تبدیل می شوند. هدف از فرآیند دگرگونی این است که پیکسل های هر زیر تصویر را تفکیک کرده یا تا آنجا که ممکن است اطلاعات را در کوچکترین تعداد ضرایب تبدیل شکل بسته بندی کند. سپس مرحله کمی سازی به صورت انتخابی ، ضرایبی را که حداقل مقدار اطلاعات را به معنای از پیش تعیین شده حمل می کنند ، حذف می کند یا درشت تر می کند (چندین روش بعداً در بخش مورد بحث قرار می گیرد). این ضرایب کوچکترین تأثیر را بر کیفیت زیر تصویر بازسازی شده دارند. فرایند رمزگذاری با کدگذاری (معمولاً با استفاده از یک کد با طول متغیر) ضرایب کوانتیزه خاتمه می یابد. هر یک یا همه مراحل رمزگذاری تبدیل را می توان با محتوای تصویر محلی که کدگذاری تبدیل تطبیقی نامیده می شود ، تطبیق داد ، یا برای همه زیر تصاویر ، کدگذاری تبدیل غیر تطبیقی ، ثابت کرد.

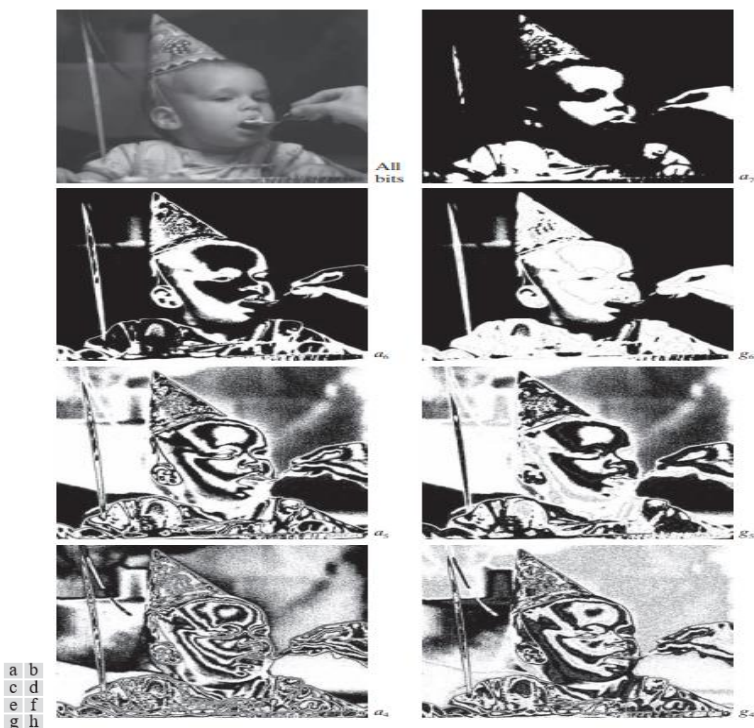
### کدگذاری تبدیل بلاک

سیستم های کدگذاری تبدیل مبتنی بر انواع تبدیل های دو بعدی گسسته ساخته شده و یا به طور گسترده مورد مطالعه قرار گرفته است. انتخاب یک تبدیل خاص در یک برنامه معین بستگی به میزان خطای بازسازی قابل

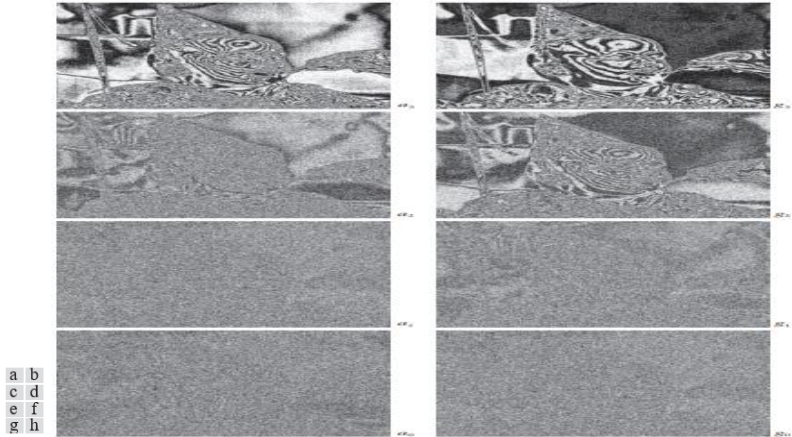
تحمل و منابع محاسباتی موجود دارد. فشرده سازی در طول کمی سازی ضرایب تبدیل (نه در مرحله تبدیل) به دست می آید.

مثال ۸.۱۳: مسدود کردن کدگذاری تبدیل با DFT، WHT و DCT.

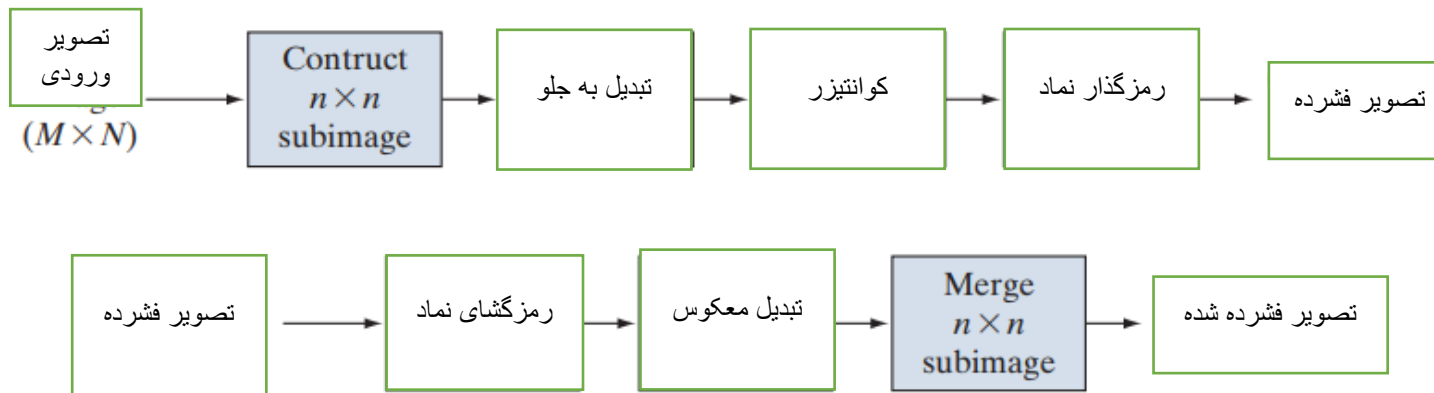
شکل های ۸.۲۲ (الف) تا (ج) سه تقریب تصویر تک رنگ ۵۱۲\*۵۱۲ را در شکل ۸.۹ (الف) نشان می دهد. این تصاویر با تقسیم تصویر اصلی به زیرصفحه هایی با ابعاد ۸ \* ۸ بدست آمده است که هر زیر تصویر را با استفاده از سه تبدیل شکل توصیف شده در فصل ۷ (تبدیل های DFT، WHT و DCT) نشان می دهد و ۵۰ درصد ضرایب حاصله را کوتاه می کند. و با استفاده از تبدیل معکوس آرایه های ضخیم .



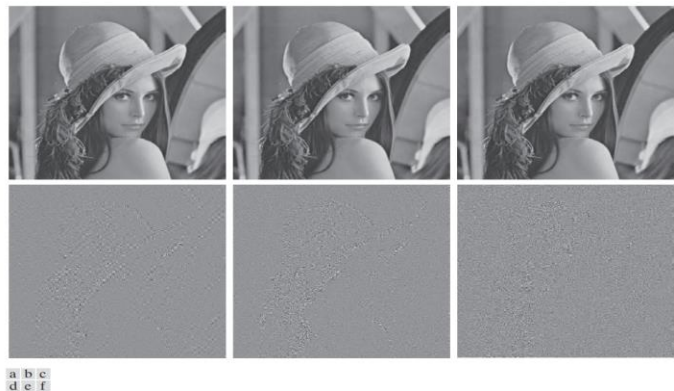
شکل ۸.۱۹ (الف) یک تصویر ۲۵۶ بیتی تک رنگ. (ب) - (ج) چهار معنی دارترین دوتایی و سطوح بیتی کد خاکستری تصویر در (a).



شکل ۸.۲۰ (الف)-(ح) چهار صفحه دوتایی کم اهمیت (ستون چپ) و کد خاکستری (ستون راست) تصویر در شکل ۸.۱۹ (الف).



شکل ۸.۲۱ سیستم کدگذاری تبدیل بلوکی: الف) رمزگذار؛ ب) رمزگشایی در هر مورد، ۳۲ ضریب حفظ شده بر اساس حداکثر بزرگی انتخاب شدند. توجه داشته باشید که در همه موارد، ۳۲ ضایعات دور ریخته شده تأثیر بصری کمی بر کیفیت تصویر بازسازی شده داشتند. با این حال، حذف آنها با خطای میانگین مربع همراه بود، که در تصاویر خطای مقیاس شده شکل ۱ مشاهده می شود. ۸.۲۲ (d) تا (f) خطاهای حقیقی rms به ترتیب ۲.۳۲، ۱.۷۸ و ۱.۱۳ شدت بودند.



شکل ۸.۲۲ تقریب های شکل ۸.۹ (الف) با استفاده از (الف) فوریه ، (ب) تبدیل والش-هادامارد و (ج) کسینوس ، همراه با تصاویر خطای مقیاس بندی شده مربوطه در (د)-(و).

تفاوت های کوچک در خطای بازسازی میانگین مربعی که در مثال قبل ذکر شد ، مستقیماً به خواص بسته بندی انرژی یا اطلاعات تبدیلیهای بکار رفته مربوط می شود. مطابق با معادله (۷-۷۵) و (۷-۷۶) بخش ۷.۵ ، یک زیر تصویر  $n * n (x, y)$  را می توان به عنوان تابعی از تبدیل ۲ بعدی  $D(u, v)$  بیان کرد:

$$\mathbf{G} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) \mathbf{S}_{uv} \quad (8-21)$$

برای  $x, y = 0, 1, 2, \dots, n-1$  ، ماتریس حاوی پیکسل های زیر تصویر ورودی ، به صراحت به عنوان یک ترکیب خطی از تصاویر پایه  $n \times n$  با اندازه  $n * n$  تعریف شده است. به یاد بیاورید که تصاویر اصلی تبدیل DFT، DCT و WHT برای  $n-1$  در شکل ها نشان داده شده است.  $7.7$  ،  $7.10$  و  $7.16$ . اگر ما در حال حاضر یک تابع پوشش ضریب تبدیل را تعریف کنیم

$$\chi(u, v) = \begin{cases} 0 & \text{if } T(u, v) \text{ satisfies a specified truncation criterion} \\ 1 & \text{otherwise} \end{cases} \quad (8-22)$$

برای  $u, v = 0, 1, 2, \dots, n-1$  ، تقریبی  $\mathbf{G}$  را می توان از انبساط کوتاه شده بدست آورد

$$\hat{\mathbf{G}} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} \chi(u, v) T(u, v) \mathbf{S}_{uv} \quad (8-23)$$

جایی که  $\chi(u, v)$  ساخته شده است تا تصاویر اصلی را که کوچکترین سهم را در مجموع معادله (۸-۲۱) دارند حذف کند. خطای میانگین مربع بین زیر-سن  $\mathbf{G}$  و تقریب  $\hat{\mathbf{G}}$  پس است

$$\begin{aligned} e_{ms} &= E \left\{ \left\| \mathbf{G} - \hat{\mathbf{G}} \right\|^2 \right\} \\ &= E \left\{ \left\| \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) \mathbf{S}_{uv} - \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} \chi(u, v) T(u, v) \mathbf{S}_{uv} \right\|^2 \right\} \\ &= E \left\{ \left\| \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) \mathbf{S}_{uv} [1 - \chi(u, v)] \right\|^2 \right\} \\ &= \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} \sigma_{T(u, v)}^2 [1 - \chi(u, v)] \end{aligned} \quad (8-24)$$

جایی که  $\|\mathbf{G} - \hat{\mathbf{G}}\|$  نرم ماتریس  $(\mathbf{G} - \hat{\mathbf{G}})$  و  $\sigma_{T(u, v)}^2$  است. در موقعیت تبدیل  $(u, v)$  ساده سازی نهایی بر اساس طبیعت معمولی تصاویر پایه و این فرض است که پیکسل های  $\mathbf{G}$  توسط یک فرایند تصادفی با میانگین صفر و کوواریانس شناخته شده ایجاد می شود. مجموع میانگین مربع خطای تقریب ، مجموع واریانس های اختلاف است. ضرایب تبدیل کارت دار ؛ یعنی ضرایبی که برای آنها  $[1 - \chi(u, v)]$  ، به طوری که  $\chi(u, v) = 0$  در معادله (۸-۲۴) ۱ است. دگرگونی هایی که بیشتر اطلاعات را در کمترین ضرایب توزیع یا بسته بندی می کنند ، بهترین تقریب های فرعی را ارائه می دهند. و در نتیجه ، کوچکترین خطاهای بازسازی. سرانجام ، تحت مفروضات که منجر به معادله



شد (۸-۲۴)، خطای میانگین مربع زیر تصویر  $MN/n^2$  یک تصویر  $M * N$  یکسان است. بنابراین خطای میانگین مربع (اندازه گیری خطای متوسط) تصویر  $M * N$  برابر خطای میانگین مربع یک زیر تصویر واحد است. مثال قبلی نشان داد که قابلیت بسته بندی اطلاعات DCT از DFT و WHT برتر است. اگرچه این حالت معمولاً برای بیشتر تصاویر صادق است، تبدیل کارهون-لوف (به فصل ۱۱ مراجعه کنید)، نه DCT، تبدیل بهینه در مفهوم بسته بندی اطلاعات است.

این به این دلیل است که KLT خطای میانگین مربع را در معادله (۸-۲۴) به حداقل می رساند. برای هر تصویر ورودی و هر تعداد ضریب حفظ شده (کرامر و ماتیوس [۱۹۵۶]). با این حال، از آنجا که KLT وابسته به داده است، بدست آوردن تصاویر پایه KLT برای هر زیر تصویر، به طور کلی، یک کار محاسباتی غیرحادی است. به همین دلیل، KLT به ندرت برای فشرده سازی تصویر استفاده می شود. در عوض، از تبدیل مانند DFT، WHT یا DCT که تصاویر اصلی آن ثابت هستند (ورودی مستقل) معمولاً استفاده می شود. از بین تبدیلات مستقل ورودی احتمالی، تبدیل های غیر سینوسی (مانند تبدیل WHT) ساده ترین پیاده سازی هستند. تبدیلات سینوسی (مانند DFT یا DCT) توانایی بسته بندی اطلاعات KLT مطلوب را نزدیک تر می کند. بنابراین، اکثر سیستم های کدگذاری متبديل شده بر اساس DCT هستند، که سازگاری خوبی بین قابلیت بسته بندی اطلاعات و پیچیدگی محاسباتی فراهم می کند. در حقیقت، خواص DCT از چنان ارزش عملی برخوردار است که DCT یک استاندارد بین المللی برای سیستم های کدگذاری تبدیل است. در مقایسه با سایر دگرگونی های مستقل ورودی، این مزایا را دارد که در یک مدار مجتمع اجرا شده است، بیشترین اطلاعات را در کمترین ضرایب  $ing$  (برای اکثر تصاویر)<sup>۳۷</sup> قرار داده و ظاهر شبیه به بلوک، به نام انسداد مصنوع را به حداقل رسانده است. هنگامی نتیجه می گیرد که مرزهای بین تصاویر فرعی قابل مشاهده باشد. این ویژگی اخیر در مقایسه با سایر تبدیلات سینوسی اهمیت ویژه ای دارد. همانطور که در شکل ۷.۱۱ (الف) بخش ۷.۶ نشان داده می شود، پرپودیتة نقطه  $n$  ضمنی DFT باعث ناپیوستگی های مرزی می شود که منجر به محتوای قابل توجه فرکانس بالا می شود. وقتی ضرایب تبدیل DFT کوتاه یا کوانتیزه می شوند، پدیده Gibbs<sup>۳۸</sup> باعث می شود که نقاط مرزی مقادیر اشتباهی به خود بگیرند، که در یک تصویر به عنوان مصنوع مسدود کننده ظاهر می شوند. به این معنا که مرزهای بین تصاویر فرعی مجاور قابل مشاهده می شوند زیرا پیکسل های مرزی تصاویر فرعی مقادیر متوسط ناپیوستگی هایی را که در نقاط مرزی شکل گرفته اند فرض می کنند [شکل ۷.۱۱ (الف) را ببینید]. DCT شکل ۷.۱۱ (b) این اثر را کاهش می دهد، زیرا دوره تناوبی  $n_2$  نقطه ای آن ذاتاً ناپیوستگی های مرزی ایجاد نمی کند.

<sup>۳۷</sup> احمد و همکاران [۱۹۷۴] برای اولین بار متوجه شد که تصاویر KLT از منبع تصویر مارکوف مرتبه اول بسیار شبیه تصاویر اصلی DCT است. با نزدیک شدن همبستگی بین پیکسل های مجاور به یک، تصاویر پایه KLT وابسته به ورودی با تصاویر پایه DCT مستقل از ورودی یکسان می شوند (کلارک [۱۹۸۵]).

<sup>۳۸</sup> این پدیده، که در اکثر متون مهندسی برق در مورد تجزیه و تحلیل مدار توضیح داده شده است، به این دلیل رخ می دهد که تبدیل فوریه قادر به همگرایی یکنواخت در ناپیوستگی ها نیست. در ناپیوستگی ها، گسترش فوریه مقادیر میانگین نقاط ناپیوستگی را می گیرد.



یک شرط اضافی برای بهینه بودن، تابع ماسک زدن است. از معادله (۸-۲۲) ضرایب حداکثر واریانس KLT را انتخاب می کند.

### انتخاب اندازه زیر تصویر

عامل مهم دیگر که بر خطای کدگذاری تبدیل و پیچیدگی محاسباتی تأثیر می گذارد، اندازه زیر تصویر است. در اکثر برنامه ها، تصاویر به صورت زیر تقسیم بندی می شوند، بنابراین همبستگی (افزونگی) بین زیر تصاویر مجاور تا حدی قابل قبول کاهش می یابد و بنابراین  $n$  یک عدد صحیح ۲ است که در آن، مانند قبل،  $n$  ابعاد زیر تصویر است. شرط اخیر، محاسبه تبدیل های زیر تصویر را ساده می کند (به روش دوبل متوالی پایه ۲ مراجعه کنید که در بخش ۴.۱۱ بحث شده است). به طور کلی، هر دو سطح فشرده سازی و پیچیدگی محاسباتی با افزایش اندازه زیر تصویر افزایش می یابد. محبوب ترین اندازه های زیر تصویر  $۸ * ۸$  و  $۱۶ * ۱۶$  هستند.

مثال ۸.۱۴: اثرات اندازه زیر تصویر بر روی کدگذاری تبدیل.

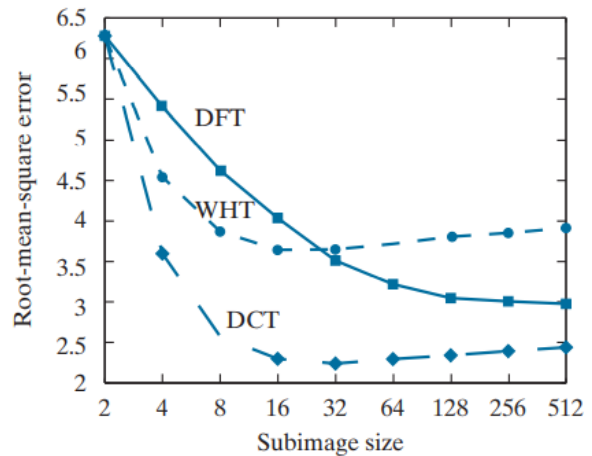
شکل ۸.۲۳ تأثیر اندازه زیر تصویر را بر روی خطای بازسازی کدگذاری تبدیل به صورت گرافیکی نشان می دهد. داده های رسم شده با تقسیم تصویر تک رنگ شکل ۸.۹ (الف) به تصاویر فرعی با اندازه  $n * n$ ، برای  $n = 2, 4, 8, 16, \dots, 256, 512$ ، محاسبه تبدیل هر زیر تصویر، کوتاه کردن ۷۵٪ از ضرایب حاصله و تبدیل معکوس آرایه های کوتاه شده استفاده می شود. توجه داشته باشید که با افزایش اندازه تصویر فرعی از  $۸ * ۸$ ، منحنی های هادامارد و کسینوس ضعیف می شوند، در حالی که خطای بازسازی فوریه همچنان در این منطقه کاهش می یابد.

با افزایش بیشتر، خطای بازسازی فوریه از منحنی والش-هادامارد عبور کرده و به نتیجه کسینوس نزدیک می شود. این نتیجه با یافته های نظری و تجربی گزارش شده توسط Netravali و Limb [۱۹۸۰] و توسط Pratt [۲۰۰۱] برای منبع تصویر دو بعدی مارکوف مطابقت دارد.

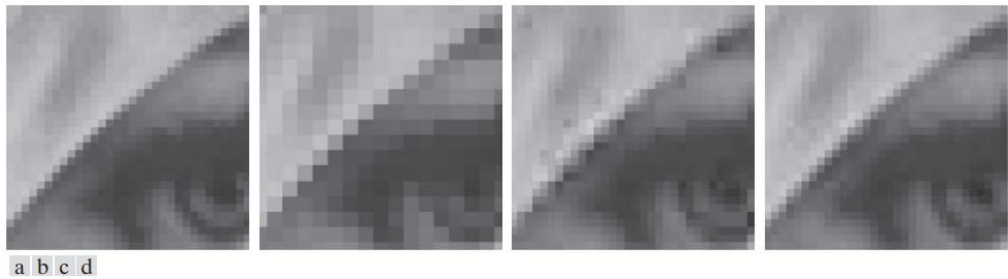
هر دو منحنی زمانی متقاطع می شوند که از ۲ تصویر فرعی  $۲ * ۲$  استفاده شود. در این مورد، تنها یکی از چهار ضرایب (۲۵٪) از هر آرایه تبدیل شده حفظ شد. ضریب اطمینان در همه موارد جزء DC بود، بنابراین تبدیل معکوس به سادگی چهار پیکسل زیر تصویر را با مقدار متوسط آنها جایگزین می کند [به معادله مراجعه کنید. (۹۲-۴)]. این وضعیت در شکل ۸.۲۴ (ب) مشهود است که بخشی از بزرگنمایی  $۲ * ۲$  نتیجه DCT را نشان می دهد. توجه داشته باشید که مصنوع مسدود کننده که در این نتیجه شایع است با افزایش اندازه زیر تصویر به  $۴ * ۴$  و  $۸ * ۸$  در شکل ها کاهش می یابد. ۸.۲۴ (ج) و (د) شکل ۸.۲۴ (الف) یک قسمت بزرگنمایی شده از تصویر اصلی را برای مرجع نشان می دهد.

### تخصیص بیت

خطای بازسازی مرتبط با گسترش سری کوتاه از معادله (۲۳-۲۸) تابعی از تعداد و اهمیت نسبی ضرایب دگرگونی است که کنار گذاشته می شوند و همچنین دقتی که برای نشان دادن ضرایب حفظ شده استفاده می شود. در اکثر سیستم های کدگذاری تبدیل ، ضرایب حفظ شده انتخاب می شوند [یعنی تابع پوشاندن معادله (۸-۲۲) بر اساس حداکثر ساخته شده است



شکل ۸.۲۳ خطای بازسازی در مقابل اندازه زیر تصویر.



شکل ۸.۲۴ تقریب های شکل ۸.۲۴ (الف) با استفاده از ۲۵٪ ضرایب DCT و (ب) زیر تصویر ۲ \* ۲ ، (ج) زیر تصویر ۴ \* ۴ ، و (د) زیر تصویر ۸ \* ۸. تصویر اصلی در (الف) یک قسمت بزرگنمایی شده از شکل ۸.۹ (الف) است.

واریانس ، که کدگذاری منطقه ای نامیده می شود یا بر اساس حداکثر اندازه ، کدگذاری آستانه نامیده می شود. به فرایند کلی کوتاه سازی ، کمی سازی و کدگذاری ضرایب یک زیر تصویر تبدیل شده معمولاً تخصیص بیت گفته می شود.

#### مثال ۸.۱۵: تخصیص بیت

شکل ۸.۲۵ (الف) و (ج) دو تقریب از شکل ۸.۹ (الف) را نشان می دهد که در آن ۸۷.۵ درصد از ضرایب DCT از هر زیر تصویر ۸ \* ۸ دور انداخته شده است. اولین نتیجه از طریق کدگذاری آستانه با حفظ هشت بزرگترین ضریب تبدیل به دست آمد و تصویر دوم با استفاده از روش کدگذاری منطقه ای ایجاد شد. در حالت دوم ، هر ضریب DCT یک متغیر تصادفی در نظر گرفته شد که توزیع آن می تواند بر روی مجموعه تمام زیر تصویرهای

تبدیل یافته محاسبه شود. هشت توزیع بزرگترین واریانس (۱۲.۵ of از ۶۴ ضرایب در تصویر فرعی ۸\*۸) قرار گرفت و برای تعیین مختصات [u و v از ضرایب ، T(u, v)] ، که برای همه حفظ شده بود ، استفاده شد. زیر تصاویر توجه داشته باشید که تصویر تفاوت آستانه شکل ۸.۲۵ (ب) دارای خطای کمتری نسبت به نتیجه کدگذاری منطقه ای در شکل ۸.۲۵ (د) است. هر دو تصویر مقیاس بندی شده اند تا خطاها بیشتر نمایان شوند. خطاهای مربوط به rms به ترتیب ۴.۵ و ۶.۵ شدت هستند.

### پیاده سازی کدگذاری منطقه ای

کدگذاری منطقه ای بر اساس مفهوم نظریه اطلاعات است که اطلاعات را به عنوان عدم قطعیت مشاهده می کند. بنابراین ، ضرایب تبدیل با حداکثر واریانس بیشترین اطلاعات تصویر را دارند و باید در فرایند کدگذاری حفظ شوند. خود واریانس ها را می توان مستقیماً از مجموعه آرایه های زیر تصویر تبدیل شده  $MN \times n^2$  (مانند مثال قبل) یا بر اساس یک مدل تصور تصور (مثلاً یک تابع خود همبستگی مارکوف) محاسبه کرد. در هر صورت ، فرایند نمونه گیری منطقه ای را می توان مطابق با معادله مشاهده کرد. (۸-۲۳) ، به عنوان ضرب هر  $T(u, v)$  در عنصر پاسخ دهنده در یک ماسک منطقه ای ، که با قرار دادن ۱ در مکانهای حداکثر واریانس و ۰ در همه مکانهای دیگر ساخته می شود. ضرایب حداکثر واریانس معمولاً در اطراف مبدأ تبدیل تصویر قرار دارند و در نتیجه ماسک منطقه ای معمولی نشان داده شده در شکل ۸.۲۶ (الف) نشان داده می شود. ضرابی که در طول نمونه گیری منطقه ای حفظ شده اند باید کوانتیزه و کدگذاری شوند ، بنابراین ماسک های منطقه ای گاهی نشان داده می شوند که تعداد بیت های مورد استفاده برای کدگذاری هر ضریب را نشان می دهد [شکل ۸.۲۶ (ب) را ببینید]. در بیشتر موارد ، به ضرایب یکسان تعداد بیت ها اختصاص داده می شود ، یا تعداد ثابت بیت ها بین آنها به طور نامساوی توزیع می شود. در مورد اول ، ضرایب معمولاً با انحرافات استاندارد عادی شده و به طور یکنواخت کوانتیزه می شوند. در حالت دوم ، یک کوانتیزر ، مانند یک کوانتیزر بهینه Lloyd-Max (به کوانتیزرهای بهینه در بخش ۸.۱۰ مراجعه کنید) ، برای هر یک از ضخامت ها طراحی شده است. برای ساختن کمیت کننده های مورد نیاز ، ضریب صفر یا DC به طور معمول با تابع چگالی ریلی مدل می شود ، در حالی که ضرایب باقی مانده با چگالی لاپلاسی یا گاوس مدل می شوند. به هر کوانتیزر ساخته شده است. متناسب با  $\log_2 s^2$ . بنابراین ، ضرایب حفظ شده در معادله<sup>۳۹</sup> (۲۳-۸) که (در زمینه بحث کنونی) بر اساس حداکثر تنوع انتخاب می شوند- متناسب با لگاریتم واریانس های ضریب ، بیت ها اختصاص داده می شوند.

<sup>۳۹</sup> همانطور که هر فاکتور ضریب ترکیبی خطی از پیکسل ها در تصویر زیر است [به معادله مراجعه کنید (۷-۲۱) ] ، نظریه محدود مرکزی نشان می دهد که با افزایش اندازه زیر تصویر ، ضرایب تمایل به گوسی شدن دارند. این نتیجه در مورد ضرایب dc صدق نمی کند ، اما تصاویر غیر منفی همیشه دارای ضرایب مثبت dc هستند.



شکل ۸.۲۵ تقریب های شکل ۸.۹ (الف) با استفاده از ۱۲.۵٪ از ضرایب DCT: (الف) - (ب) نتایج کدگذاری آستانه. (ج) - (د) نتایج کدگذاری منطقه ای. مقیاس تصاویر مختلف ۴ است.

### پیاده سازی آستانه کدگذاری

برنامه نویسی منطقه ای معمولاً با استفاده از یک ماسک ثابت برای همه زیر تصاویر اجرا می شود. با این حال ، کدگذاری آستانه ذاتاً سازگار است به این معنا که محل ضرایب تبدیل شکل که برای هر زیر تصویر حفظ می شود ، از یک زیر تصویر به تصویر دیگر متفاوت است. در واقع ، کدگذاری آستانه ، رویکرد کدگذاری تبدیل تطبیقی است که اغلب به دلیل سادگی محاسباتی در عمل مورد استفاده قرار می گیرد. این مفهوم اساسی این است که ، برای هر زیر تصویر ، ضرایب دگرگونی با بزرگترین بزرگی ، مهمترین سهم را در بازسازی کیفیت زیر تصویر دارند ، همانطور که در آخرین مثال نشان داده شد.

1	1	1	1	1	0	0	0	8	7	6	4	3	2	1	0
1	1	1	1	0	0	0	0	7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	0	6	5	4	3	3	1	1	0
1	1	0	0	0	0	0	0	4	4	3	3	2	1	0	0
1	0	0	0	0	0	0	0	3	3	3	2	1	1	0	0
0	0	0	0	0	0	0	0	2	2	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	1	5	6	14	15	27	28
1	1	1	0	0	0	0	0	2	4	7	13	16	26	29	42
1	1	0	0	0	0	0	0	3	8	12	17	25	30	41	43
1	0	0	0	0	0	0	0	9	11	18	24	31	40	44	53
0	0	0	0	0	0	0	0	10	19	23	32	39	45	52	54
0	0	0	0	0	0	0	0	20	22	33	38	46	51	55	60
0	0	0	0	0	0	0	0	21	34	37	47	50	56	59	61
0	0	0	0	0	0	0	0	35	36	48	49	57	58	62	63

شکل ۸.۲۶ (الف) ماسک منطقه ای ، (ب) تخصیص بیت منطقه ای ، (ج) ماسک آستانه ، و (د) ترتیب ضرایب آستانه. سایه زدن ، ضرایب حفظ شده را برجسته می کند.

زیرا مکان حداکثر ضرایب متفاوت است. یک تصویر فرعی به دیگری ، عناصر  $\log_2 \sigma_{T(u,v)}^2$  معمولاً دوباره مرتب می شوند (در روش از پیش تعیین شده) برای تشکیل یک دنباله کدگذاری ۱ بعدی و طول اجرا. شکل ۸.۲۶ (ج) یک

ماسک آستانه معمولی را برای یک زیر تصویر از یک تصویر فرضی نشان می دهد. این ماسک راهی مناسب برای تجسم فرایند کدگذاری آستانه برای زیر تصویر مربوطه و همچنین توصیف ریاضی این فرایند با استفاده از معادله (۸-۲۳) را ارائه می دهد.

هنگامی که ماسک اعمال می شود [از طریق معادله (۸-۲۳)] به زیر تصویری که برای آن مشتق شده است ، و آرایه  $n * n$  به دست آمده مجدداً ترتیب می شود تا مطابق با الگوی ترتیب زیگزگی شکل ۸.۲۶ (d) ترتیب متوالی  $n \times n$  عنصر را شکل دهد ، ترتیب مجدد ۱ دنباله  $D$  شامل چندین دور طولانی ۰ می باشد. [الگوی زیگزگ با شروع از ۰ در شکل ۸.۲۶ (d) و دنبال کردن اعداد به ترتیب آشکار می شود.] این دویدن ها معمولاً در طول اجرا کدگذاری می شوند. ضرایب غیر صفر یا حفظ شده ، مربوط به محل های ماسک که حاوی ۱ هستند ، با استفاده از یک کد با طول متغیر نشان داده می شوند.

سه راه اساسی برای آستانه یک تصویر فرعی تبدیل یافته یا به صورت متفاوت ، ایجاد یک تابع پوشاندن آستانه subimage از فرم ارائه شده در معادله وجود دارد. (۸-۲۲): (۱) یک آستانه جهانی واحد را می توان برای همه تصاویر فرعی اعمال کرد. (۲) برای هر زیر تصویر می توان از آستانه متفاوتی استفاده کرد ، یا ؛ (۳) آستانه را می توان به عنوان تابعی از مکان هر ضریب در زیر تصویر تبدیل داد. در رویکرد اول ، میزان فشرده سازی بسته به تعداد ضرایب فراتر از آستانه جهانی ، از تصویر به تصویر دیگر متفاوت است. در مورد دوم ، که بزرگترین کدگذاری نامیده می شود ، برای هر زیر تصویر ، تعداد مشابهی از ضرایب کنار گذاشته می شود. در نتیجه ، نرخ کد ثابت و از قبل مشخص است. تکنیک سوم ، مانند روش اول ، دارای نرخ کد متغیر است ، اما مزیتی را ارائه می دهد که آستانه و کمی سازی می تواند با جایگزینی  $\chi(u,v)T(u,v)$  در معادله (۸-۲۳) ترکیب شود. با

$$\hat{T}(u,v) = \text{round} \left[ \frac{T(u,v)}{Z(u,v)} \right] \quad (8-25)$$

جایی که  $T(u,v)$  تقریبی آستانه ای و کمی شده از  $T(u,v)$  است و  $Z(u,v)$  عنصری از آرایه عادی سازی تبدیل زیر است:

$$\mathbf{Z} = \begin{bmatrix} Z(0,0) & Z(0,1) & \dots & Z(0,n-1) \\ Z(1,0) & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ Z(n-1,0) & Z(n-1,1) & \dots & Z(n-1,n-1) \end{bmatrix} \quad (8-26)$$

$N$  در "بزرگترین کدگذاری" یک بعد تصویر نیست ، بلکه به تعداد ضرایب نگهداری شده اشاره دارد. قبل از تبدیل عادی (آستانه و کوانتیزه) زیر تصویر ،  $T(u,v)$  ، می توان معکوس تبدیل کرد تا تقریبی از زیر تصویر  $g(x,y)$  بدست آورد ، باید

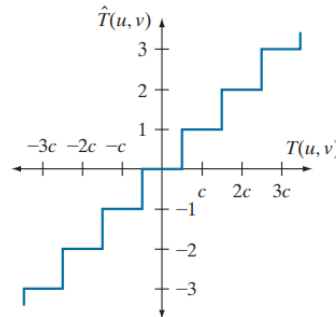
در  $Z(u, v)$  ضرب شود. آرایه غیر عادی حاصل شده، نشان دهنده  $\hat{T}(u, v)$ ،  $\hat{T}(u, v)$  یک است تقریب  $\hat{T}(u, v)$ .

$$\hat{T}(u, v) = \hat{T}(u, v)Z(u, v) \quad (8-27)$$

تبدیل معکوس  $\hat{T}(u, v)$ ، تقریب فرعی را از حالت فشرده خارج می کند. شکل ۸.۲۷ (الف) معادله (۸-۲۵) را به صورت گرافیکی نشان می دهد. برای موردی که  $Z(u, v)$  در آن است. مقدار خاصی به آن اختصاص داده C. توجه داشته باشید که  $\hat{T}(u, v)$  مقدار صحیح k را در صورت و فقط اگر فرض می کند

$$kc - \frac{c}{2} \leq T(u, v) < kc + \frac{c}{2} \quad (8-28)$$

اگر  $Z(u, v) > 2T(u, v)$  سپس  $T(u, v) = 0$  و ضریب تبدیل کاملاً کوتاه یا دور انداخته می شود. وقتی  $T(u, v)$  با یک کد طول متغیر نشان داده می شود که با افزایش اندازه k طول آن افزایش می یابد، تعداد بیت هایی که برای نشان دادن  $T(u, v)$  استفاده می شود با مقدار C کنترل می شود. بنابراین، عناصر Z را می توان برای دستیابی به سطوح مختلف فشرده سازی مقیاس بندی کرد. شکل ۸.۲۷ (ب) یک آرایه معمولی معمولی را نشان می دهد. این آرایه، که به طور گسترده در تلاشهای استانداردسازی JPEG استفاده شده است (بخش بعدی را ببینید)، هر ضریب یک تصویر فرعی تبدیل یافته را با توجه به اهمیت ادراکی یا روانی - بصری تعیین شده از لحاظ شهوانی، وزن می کند.



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

شکل ۸.۲۷ (الف) منحنی کمی سازی آستانه [معادله را ببینید. (۸-۲۸)]. (ب) یک ماتریس نرمال سازی معمولی.

### مثال ۸.۱۶: تصویر کدگذاری آستانه.

شکل ۸.۲۸ (الف) تا (و) شش تقریب کدگذاری شده از آستانه از تصویر تک رنگ را در شکل ۸.۹ (الف) نشان می دهد. همه تصاویر با استفاده از DCT ۸ ۸ ۸ و آرایه نرمال سازی شکل ۸.۲۷ (ب) تولید شده اند. اولین نتیجه، که نسبت فشرده سازی حدود ۱۲ به ۱ (یعنی  $C = 12$ ) را فراهم می کند، با استفاده مستقیم از آن آرایه نرمال سازی بدست آمد. نتایج باقی مانده، که تصویر اصلی را ۱۹، ۳۰، ۴۹، ۸۵ و ۱۸۲ به ۱ فشرده می کند، پس از ضرب (مقیاس بندی) آرایه های نرمال سازی در ۲، ۴، ۸، ایجاد شد. به ترتیب ۱۶ و ۳۲ خطاهای مربوط به rms عبارتند از ۳.۸۳، ۴.۹۳، ۶.۶۲، ۹.۳۵، ۱۳.۹۴ و ۲۲.۴۶ سطح شدت.

## JPEG

یکی از معروفترین و جامع ترین استانداردهای تراکم پیوسته ، استاندارد فرمت JPEG است. این سه سیستم کدگذاری مختلف را تعریف می کند: (۱) یک سیستم برنامه نویسی پایه زیان آور ، که بر اساس DCT است و برای اکثر برنامه های فشرده سازی مناسب است. (۲) یک سیستم کدگذاری گسترده برای فشرده سازی بیشتر ، دقت بیشتر یا برنامه های بازسازی پیشرونده. و (۳) یک سیستم کدگذاری مستقل بدون اتلاف برای فشرده سازی برگشت پذیر. برای سازگاری با JPEG ، یک محصول یا سیستم باید شامل پشتیبانی از سیستم پایه باشد. هیچ قالب فایل خاص ، وضوح فضایی یا مدل فضای رنگ مشخص نشده است.

در سیستم پایه ، که اغلب به آن سیستم پایه پی در پی گفته می شود ، دقت داده های ورودی و خروجی به ۸ بیت محدود می شود ، در حالی که مقادیر کوانتیزه DCT به ۱۱ بیت محدود می شود. فشرده سازی به خودی خود در سه مرحله متوالی انجام می شود: محاسبه DCT ، کمی سازی و تعیین کد متغیر. تصویر برای اولین بار به بلوک های پیکسل اندازه  $8 * 8$  تقسیم می شود که از چپ به راست ، از بالا به پایین پردازش می شوند. با مواجهه با هر بلوک  $8 * 8 * 8$  یا زیر تصویر ، ۶۴ پیکسل آن با تفریق مقدار  $k=12$  ، که حداکثر تعداد سطوح شدت است ، تبدیل سطح می دهند. سپس تبدیل کسینوس گسسته ۲ بعدی بلوک محاسبه می شود و مطابق با (۸-۲۵) ، معیار اندازه گیری می شود. و مجدداً با استفاده از الگوی زیگزاگ شکل ۸.۲۶ (d) ، مجدداً مرتب شده ، تا یک توالی یک بعدی از ضرایب کوانتیزه تشکیل شود.

از آنجا که آرایه یک بعدی مرتب شده تحت الگوی زیگزاگ شکل ۸.۲۶ (d) از نظر کیفی با توجه به افزایش فرکانس فضایی مرتب شده است ، روش کدگذاری JPEG طوری طراحی شده است که از مزایای طولانی صفرها که به طور معمول ناشی از مرتب سازی مجدد به طور خاص ، ضرایب  $AC^{40}$  غیر صفر با استفاده از یک کد طول متغیر کد گذاری می شوند که مقادیر ضریب و تعداد صفرهای قبلی را مشخص می کند. ضرایب DC تفاوت نسبت به ضریب DC زیر تصویر قبلی دارد. کدهای JPEG Huffman پیش فرض برای روشنایی یک تصویر رنگی یا شدت یک تصویر تک رنگ در وب سایت کتاب موجود است. آرایه کمیت روشنایی توصیه شده توسط JPEG در شکل ۸.۲۷ (b) آورده شده است و می تواند برای ایجاد سطوح مختلف فشرده سازی مقیاس بندی شود. مقیاس بندی این آرایه به کاربران امکان می دهد "کیفیت" فرمت های JPEG را انتخاب کنند. اگرچه جداول کدگذاری پیش فرض و آرایه های کمی سازی هم برای پردازش رنگی و هم تک رنگ ارائه می شود ، کاربر مختار است جدولها و/یا آرایه های سفارشی بسازد ، که ممکن است با ویژگیهای تصویر فشرده شده سازگار باشد.

برای جداول کد پیش فرض JPEG Huffman به وب سایت کتاب مراجعه کنید: (۱) یک جدول دسته بندی JPEG (۲) یک جدول کد پیش فرض DC و (۳) یک جدول کد پیش فرض AC.

<sup>۴۰</sup> در استاندارد ، اصطلاح AC تمام ضرایب دگرگونی را به استثنای ضریب صفر یا DC نشان می دهد.



شکل ۸.۲۸ تقریب های شکل ۸.۹ (الف) با استفاده از آرایه DCT و نرمال سازی شکل ۸.۲۷ (ب): (الف) Z ، (ب)

Z۳۲ ، Z۲ (ج) Z۴ ، (د) Z۸ ، Z۱۶ (e) ، و (و) Z۳۲

مثال ۸.۱۷: کدگذاری و رمزگذاری پایه JPEG

فشرده سازی و بازسازی زیر تصویر ۸ \* ۸ زیر را با استاندارد پایه JPEG در نظر بگیرید:

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

تصویر اصلی از ۲۵۶ یا ۲۸ شدت احتمالی تشکیل شده است ، بنابراین فرایند کدگذاری با تبدیل سطح پیکسل های زیر تصویر اصلی به میزان  $2^{-7}$  یا  $2^{-8}$  شدت آغاز می شود. آرایه تبدیل یافته حاصل است

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

که وقتی مطابق با DCT جلو معادله (۷-۳۱) تبدیل می شود. با  $r(x, y, u, v) = s(x, y, u, v)$  از معادله (۷-۷)

(۸۵) برای  $n = ۸$  می شود



-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

اگر JPEG از آرایه عادی سازی شکل ۸.۲۷ (b) برای کمی سازی آرایه تبدیل شده استفاده می کند ، مقیاس بندی شده و کوتاه می شود [یعنی مطابق با معیار نرمال می شود. (۸-۲۵)] هستند

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

جایی که ، برای مثال ، ضریب DC به عنوان محاسبه می شود

$$\hat{T}(0,0) = \text{round} \left[ \frac{T(0,0)}{Z(0,0)} \right]$$

$$= \text{round} \left[ \frac{-415}{16} \right] = -26$$

جایی که ، برای مثال ، ضریب DC به عنوان محاسبه می شود

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

جایی که نماد EOB نشان دهنده وضعیت پایان بلوک است. یک کلمه کد ویژه EOB Huffman (دسته ۰ و طول اجرا ۰ جدول کد پیش فرض JPEG JPEG را در وب سایت کتاب ببینید) ارائه شده است تا نشان دهد که بقیه ضرایب در یک ترتیب مرتب شده صفر هستند.

ساخت کد JPEG پیش فرض برای ترتیب مجدد ضرایب ضریب با محاسبه تفاوت بین ضرایب DC فعلی و زیر تصویر رمزگذاری شده قبلی شروع می شود. با فرض ضرایب DC زیر تصویر تبدیل یافته و کمی شده در سمت چپ فوری آن ۱۷ - بود ، تفاوت DPCM حاصل  $[-26 - (-17)]$  یا ۹- است که در رده تفاوت ۴ DC قرار دارد از جدول دسته بندی ضرایب JPEG (به وب سایت کتاب مراجعه کنید). مطابق با هافمن پیش فرض کد تفاوت ، کد پایه مناسب برای تفاوت دسته ۴ ۱۰۱ است (یک کد ۳ بیتی) ، در حالی که طول کل ضریب رده ۴ کاملاً رمزگذاری شده ۷ بیت است. ۴ بیت باقیمانده باید از حداقل بیت‌های قابل توجه (LSB) مقدار تفاوت ایجاد شود. برای یک دسته تفاوت کلی DC (مثلاً دسته K) ، یک بیت K اضافی مورد نیاز است و به عنوان K LSB تفاوت مثبت یا K

LSB اختلاف منفی ۱ محاسبه می شود. برای تفاوت ۹-، LSB های مناسب هستند. ۱۱۱۰۱۱۱ - یا ۰۱۱۰، و کلمه کد DC DPCM کامل ۱۰۱۰۱۱۰ است. ضرایب AC غیر صفر آرایه مرتب شده به طور مشابه کدگذاری می شوند. تفاوت اصلی در این است که هر کلمه کد پیش فرض AC هافمن به تعداد ضرایب صفر ارزشی که قبل از کدگذاری ضریب صفر و همچنین رده بزرگی ضرایب غیر صفر بستگی دارد، بستگی دارد. (ستون با برچسب Run/Category را در جدول کد JPEG AC در وب سایت کتاب مشاهده کنید). بنابراین، اولین ضریب AC بدون صفر AC از آرایه مرتب شده (۳-) به عنوان ۰۱۰۰ کدگذاری شده است. اولین ۲ بیت این کد نشان می دهد که ضریب فاضلاب در رده ۲ بوده و قبل از آن هیچ ضرایب ارزشی صفر وجود نداشته است. ۲ بیت آخر با همان فرایندی که برای رسیدن به LSB های کد تفاوت DC استفاده می شود، تولید می شود. با ادامه این روش، آرایه کاملاً کدگذاری شده (دوباره مرتب شده) است

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001  
001 100101 11100110 110110 0110 11110100 000 1010

جایی که فضاها فقط برای خوانایی درج شده اند. اگرچه در این مثال نیازی به آن نبود، اما کد پیش فرض JPEG حاوی یک کلمه کد ویژه برای اجرای ۱۵ صفر و به دنبال آن صفر است. تعداد کل بیت ها در آرایه کاملاً مرتب شده مجدد (و بنابراین تعداد بیت های مورد نیاز برای نمایش کل زیر تصویر ۸ \* ۸، ۸ بیتی این مثال) ۹۲ است. نسبت فشرده سازی حاصله ۵۱۲ ۹۲ یا حدود ۵.۶:۱ برای فشرده سازی یک زیر تصویر فشرده JPEG، رمزگشا باید ابتدا ضرایب ترانس شکل نرمال شده را که منجر به جریان بیت فشرده شده است، افزودگی کند. از آنجا که یک دنباله دوتایی با کد هافمن فوری و به طور منحصر به فرد رمزگشایی می شود، این مرحله به راحتی با روش جدول جستجوی ساده انجام می شود. در اینجا، آرایه بازسازی شده از فاکتورهای کمی شده است

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

پس از عادی سازی مطابق با معادله (۸-۲۷)، آرایه می شود

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

که در آن، برای مثال، ضرایب DC به عنوان محاسبه می شود

$$\hat{T}(0,0) = \hat{T}(0,0)Z(0,0) = (-26)(16) = -416$$

زیر تصویر کاملاً بازسازی شده با در نظر گرفتن DCT معکوس آرایه غیر عادی مطابق با معادلات (۷-۳۲) و (۷-۸۵) بدست می آید. برای به دست آوردن

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59
-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44

و تبدیل سطح هر پیکسل معکوس تبدیل شده توسط ۲۷ (یا ۲۸-) به تابع

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

هرگونه تفاوت بین زیر تصویر اصلی و بازسازی شده ناشی از ماهیت از دست دادن فرآیند فشرده سازی و رفع فشار JPEG است. در این مثال ، خطاها از ۱۴- تا ۱۱- متغیر هستند و به شرح زیر توزیع می شوند:

-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	6	4	3	-2
3	8	4	-4	2	6	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

خطای ریشه میانگین مربع کل فرآیند فشرده سازی و بازسازی تقریباً است

مثال ۸.۱۸: تصویر برنامه نویسی JPEG

شکل ۸.۲۹ (a) و (d) دو تقریب JPEG از تصویر تک رنگ را در شکل ۸.۹ (a) نشان می دهد. اولین نتیجه فشرده سازی ۲۵:۱ است. دومی تصویر اصلی را ۵۲:۱ فشرده می کند. تفاوت بین تصویر اصلی و تصاویر بازسازی شده در شکل ۸.۲۹ (الف) و (د) در شکل ۸.۲۹ (b) و (e) . به ترتیب نشان داده شده است. خطاهای مربوط به ۵٫۴ rms و ۱۰٫۷ شدت هستند. خطاها در تصاویر بزرگنمایی شده در شکل ها به وضوح قابل مشاهده است. ۸.۲۹ (ج) و (و). این تصاویر بخش بزرگی از شکل ها را نشان می دهد. ۸.۲۹ (a) و (d) به ترتیب. توجه داشته باشید که مصنوع مسدود کننده JPEG با فشرده سازی افزایش می یابد.

## ۸.۱۰ کدگذاری پیش بینی کننده

ما اکنون به رویکرد ساده تری روی می آوریم که فشرده سازی خوبی را بدون سربار محاسباتی قابل توجه انجام می دهد و می تواند عاری از خطا یا ضرر باشد. این رویکرد، که معمولاً به عنوان کدگذاری پیش بینی کننده شناخته می شود، مبتنی بر حذف اضافات پیکسل های فاصله دار - در فضا و/یا زمان - با استخراج و کدگذاری فقط اطلاعات جدید در هر پیکسل است. اطلاعات جدید یک پیکسل به عنوان تفاوت بین مقدار حقیقی و پیش بینی شده پیکسل تعریف می شود.

با ارجاع به جداول ۸.۳-۸.۵، کدگذاری پیش بینی در استفاده می شود •JBIG۲ •JPEG •MPEG-LS •VC-۱ و سایر استانداردها و قالب های فایل فشرده سازی. H.۲۶۱، H.۲۶۲، H.۲۶۳ و H.۲۶۴ •HDV •VC-۱ و سایر استانداردها و قالب های فایل فشرده سازی.

## ۸.۱۰ کدگذاری پیش بینی بدون عارضه

شکل ۸.۳۰ اجزای اساسی یک سیستم کدگذاری پیش بینی بدون ضرر را نشان می دهد. این سیستم از رمزگذار و رمزگشایی تشکیل شده است که هر یک حاوی یک پیش بینی کننده یکسان هستند. به عنوان نمونه های پی در پی سیگنال ورودی زمان گسسته،  $f(n)$ ، به رمزگذار معرفی می شود، پیش بینی کننده مقدار پیش بینی شده هر نمونه را بر اساس تعداد مشخصی از نمونه های قبلی تولید می کند. سپس خروجی پیش بینی کننده به نزدیکترین عدد صحیح گرد می شود، با  $\hat{f}(n)$  نشان داده می شود و برای ایجاد تفاوت یا خطای پیش بینی استفاده می شود.

$$e(n) = f(n) - \hat{f}(n) \quad (8-29)$$

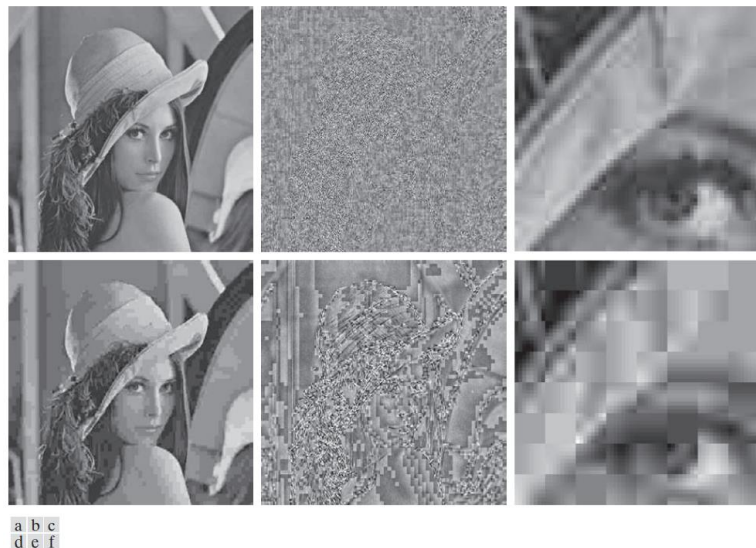
که با استفاده از یک کد طول متغیر (توسط رمزگذار نماد) برای ایجاد عنصر بعدی جریان داده فشرده کدگذاری می شود. رمزگشای شکل ۸.۳۰ (ب)  $e(n)$  را از کلمات رمز دریافت شده با طول متغیر بازسازی کرده و عمل معکوس را انجام می دهد.

$$f(n) = e(n) + \hat{f}(n) \quad (8-30)$$

برای فشرده سازی یا افزونگی دنباله ورودی اصلی روشهای مختلف محلی، جهانی و تطبیقی (به بخش بعدی بعنوان کدگذاری پیش بینی کننده از دست رفته مراجعه کنید) می توانید برای تولید  $f(n)$  استفاده کنید. در بسیاری از موارد، پیش بینی به عنوان ترکیبی خطی از  $m$  نمونه های قبلی شکل می گیرد. به این معنا که،

$$\hat{f}(n) = \text{round} \left[ \sum_{i=1}^m \alpha_i f(n-i) \right] \quad (8-31)$$

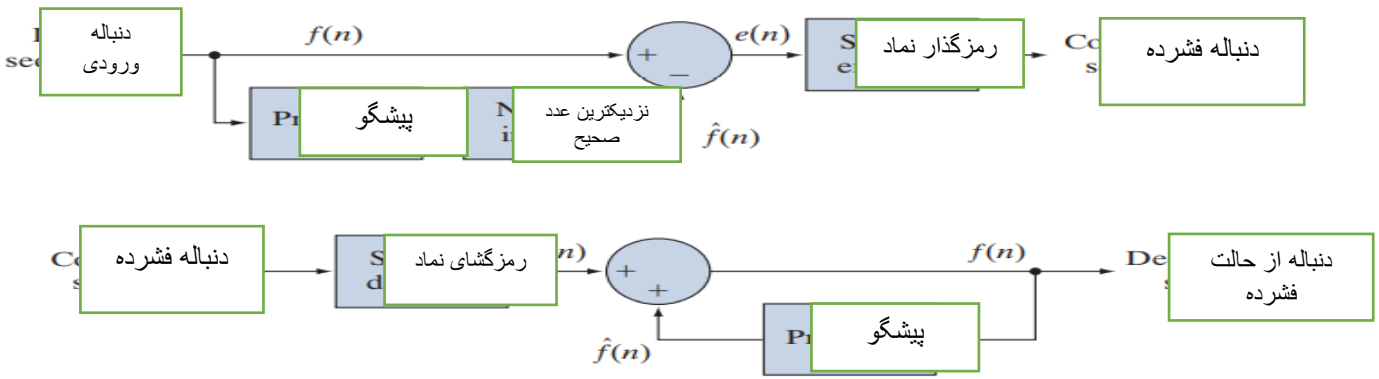
جایی که  $m$  ترتیب پیش بینی خطی است ،  $\text{round}$  تابع است که برای گرد کردن یا نزدیکترین عمل صحیح به کار می رود و  $ai$  برای  $i = 1, 2, \dots, m$  پیش بینی است



شکل ۸.۲۹ دو تقریب JPEG از شکل ۸.۹ (الف). هر سطر حاوی یک نتیجه پس از فشرده سازی و بازسازی ، تفاوت مقیاس شده بین نتیجه و تصویر اصلی و یک قسمت بزرگنمایی شده از تصویر بازسازی شده است. ضرایب اگر توالی ورودی در شکل ۸.۳۰ (الف) نمونه های یک تصویر در نظر گرفته شود ،  $f(n)$  در معادله (۸-۲۹) تا (۸-۳۱) پیکسل هستند و  $m$  نمونه هایی که برای پیش بینی مقدار هر پیکسل استفاده می شود از خط اسکن فعلی (به نام کدگذاری پیش بینی گوش  $D-1$ ) از اسکن فعلی و قبلی می آید. خطوط (کدگذاری پیش بینی خطی دو بعدی) ، یا از تصویر فعلی و تصاویر قبلی در دنباله ای از تصاویر (کدگذاری پیش بینی خطی سه بعدی نامیده می شود). بنابراین ، برای کدگذاری تصویر پیش بینی خطی ۱ بعدی ، معادله (۸-۳۱) را می توان به صورت نوشت

$$\hat{f}(x, y) = \text{round} \left[ \sum_{i=1}^m \alpha_i f(x, y - i) \right] \quad (8-32)$$

جایی که هر نمونه اکنون صریحاً به عنوان تابعی از مختصات فضایی تصویر ورودی ،  $x$  و  $y$  بیان شده است. توجه داشته باشید که معادله (۸-۳۲) نشان می دهد که پیش بینی خطی ۱ بعدی تابعی از پیکسل های قبلی به تنهایی در خط فعلی است. در کدگذاری پیش بینی دو بعدی ، پیش بینی تابعی از پیکسل های قبلی در یک اسکن چپ به راست ، از بالا به پایین یک تصویر است. در حالت سه بعدی ، بر اساس این پیکسل ها و پیکسل های قبلی فریم های قبلی است. معادله (۳۲-۳۸) را نمی توان برای اولین متر پیکسل هر خط ارزیابی کرد ، بنابراین این پیکسل ها باید با استفاده از وسایل دیگر (مانند کد هافمن) کدگذاری شوند و به عنوان سربار فرایند کدگذاری پیش بینی کننده در نظر گرفته شوند. نظرات مشابه در مورد موارد با ابعاد بالاتر اعمال می شود.



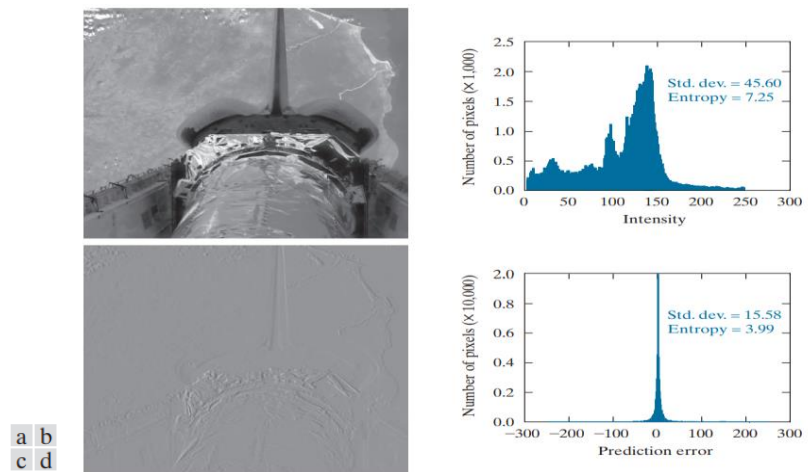
تصویر ۸.۳۰ مدل کدگذاری پیش بینی بدون ضرر: الف) رمزگذار ؛ ب) رمزگشایی  
 مثال ۸.۱۹: کدگذاری پیش بینی کننده و افزودگی فضایی.

کدگذاری تصویر تک رنگ شکل ۸.۳۱ (الف) را با استفاده از پیش بینی خطی ساده مرتبه اول (یعنی  $m = 1$ ) از معادله (۸-۳۲) در نظر بگیرید.

$$\hat{f}(x, y) = \text{round}[af(x, y - 1)] \quad (8-33)$$

این معادله یک ساده سازی معادله (۸-۳۲) ، است. با  $m = 1$  و زیرنویس ضریب پیش بینی تنها  $a$  به عنوان غیر ضروری حذف شده است. پیش بینی کننده این شکل کلی پیش بینی کننده پیکسل قبلی نامیده می شود و روش کدگذاری پیش بینی کننده مربوطه به عنوان کدگذاری دیفرانسیل یا کدگذاری پیکسلی قبلی شناخته می شود. شکل (c) ۸,۳۱ تصویر خطای پیش بینی  $e(x, y) = f(x, y) - \hat{f}(x, y)$  را نشان می دهد که از معادله حاصل می شود. (۸-۳۳) با  $a \approx 1$  مقیاس بندی این تصویر به گونه ای است که شدت ۱۲۸ نشان دهنده خطای پیش بینی صفر است در حالی که همه خطاهای پیش بینی مثبت و منفی غیر صفر (برآورد زیر و بیشتر) به ترتیب به عنوان سایه روشن تر و تیره تر خاکستری نمایش داده می شوند. میانگین مقدار تصویر پیش بینی ۱۲۸.۲۶.۲۶ است. از آنجا که شدت ۱۲۸ مطابق با خطای پیش بینی ۰ است ، متوسط خطای پیش بینی تنها ۰.۲۶ بیت است. شکل ۸.۳۱ (ب) و (د) هیستوگرام شدت تصویر را در شکل ۸.۳۱ (الف) و هیستوگرام خطای پیش بینی  $e(x, y)$  (به ترتیب نشان می دهد. توجه داشته باشید که انحراف استاندارد خطای پیش بینی در شکل ۸.۳۱ (د) بسیار کوچکتر از انحراف استاندارد شدت ها در تصویر اصلی است. علاوه بر این ، آنتروپی خطای پیش بینی ، که با استفاده از معادله برآورد شده است. (۸-۷) ، به میزان قابل توجهی کمتر از آنتروپی تخمینی تصویر اصلی است (۳.۹۹ بیت پیکسل در مقابل ۷.۲۵ بیت پیکسل). این کاهش در آنتروپی نشان دهنده حذف بسیاری از افزودگی فضایی است ، علیرغم این واقعیت که برای تصاویر  $k$ -bit  $(k-1)$ -bit اعداد برای نشان دادن دقیق توالی خطای پیش بینی  $e(x, y)$  مورد نیاز است. (توجه داشته باشید که خطای پیش بینی رمزگذاری شده با طول متغیر ، تصویر فشرده شده است.) به طور کلی ، حداکثر فشار یک روش کدگذاری پیش بینی کننده را می توان با تقسیم متوسط تعداد بیت های مورد استفاده برآورد کرد. نشان دادن هر پیکسل در تصویر اصلی با برآورد آنتروپی خطای

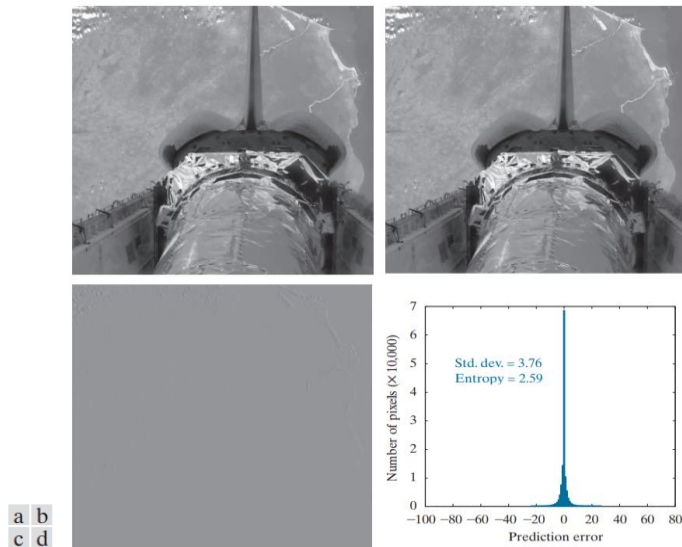
پیش بینی. در این مثال ، از هر روش کدگذاری با طول متغیر می توان برای کدگذاری  $(x, y)$  استفاده کرد ، اما تراکم به دست آمده به حدود ۸.۳۹۹ ، یا ۲:۱ محدود می شود.



شکل ۸.۳۱ (الف) نمایی از زمین از شاتل فضایی در حال گردش. (ب) هیستوگرام شدت (الف). (ج) تصویر خطای پیش بینی ناشی از معادله (۸-۳۳). (د) هیستوگرام خطای پیش بینی. (تصویر اصلی از ناسا گرفته شده است). مثال قبل نشان می دهد که فشردن سازی به دست آمده در کدگذاری پیش بینی مستقیماً به کاهش آنتروپی مربوط می شود که از نگاشتن یک تصویر ورودی در یک دنباله خطای پیش بینی ، که اغلب باقیمانده پیش بینی نامیده می شود ، مرتبط است. از آنجا که افزونگی فضایی با فرایند پیش بینی و تفاضل حذف می شود ، تابع چگالی احتمال باقی مانده پیش بینی به طور کلی بسیار صفر است و با واریانس نسبتاً کوچک (در مقایسه با توزیع شدت ورودی) مشخص می شود. به در واقع ، اغلب با یک صفر PDF غیرمرتبط لاپلاسیان مدل سازی می شود

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{\frac{-\sqrt{2}|e|}{\sigma_e}} \quad (8-34)$$

جایی که  $se$  انحراف استاندارد  $e$  است.



شکل ۸.۳۲ (الف) و (ب) دو نمای زمین از فیلم شاتل فضایی در حال گردش. (ج) تصویر خطای پیش بینی ناشی از معادله (۸-۳۵). (د) هیستوگرام خطای پیش بینی. (تصاویر اصلی از ناسا گرفته شده است).  
مثال ۸.۲۰: کدگذاری پیش بینی کننده و افزونگی زمانی.

تصویر شکل ۸.۳۱ (الف) قسمتی از یک فرمت ویدئویی ناسا است که در آن زمین با توجه به یک دوربین ثابت متصل به شاتل فضایی از چپ به راست حرکت می کند. در شکل ۸.۳۲ (ب) همراه با قاب بلافاصله قبلی آن در شکل ۸.۳۲ (الف) تکرار می شود. استفاده از پیش بینی خطی مرتبه اول

$$\hat{f}(x, y, t) = \text{round}[\alpha f(x, y, t - 1)] \quad (8-35)$$

با  $\alpha = 1$ ، شدت پیکسل های شکل ۸.۳۲ (ب) را می توان از پیکسل های مربوطه در (الف) پیش بینی کرد. شکل ۸.۳۴ (ج) تصویر پیش بینی شده باقی مانده  $e(x, y, t) = f(x, y, t) - \hat{f}(x, y, t)$  است. شکل ۸.۳۱ (د) هیستوگرام  $e(x, y, t)$  است. توجه داشته باشید که خطای پیش بینی بسیار کمی وجود دارد. انحراف استاندارد خطا بسیار کوچکتر از مثال قبلی است:  $3.76$  بیت پیکسل در مقابل  $15.58$  بیت پیکسل. علاوه بر این، آنتروپی خطای پیش بینی [با استفاده از معادله محاسبه می شود. (۸-۷)] کاهش یافته است از پیکسل  $3.99$  تا  $2.59$  بیت (مجدداً به یاد بیاورید که خطای پیش بینی رمزگذاری شده با طول متغیر، تصویر فشرده شده است.) با کدگذاری طول متغیر پیش بینی باقیمانده، تصویر اصلی تقریباً  $2.59$  یا  $3.1$ : $1$  فشرده می شود، که  $50$  پیشرفت نسبت به  $2$ : $1$  بهبود یافته است. فشرده سازی با استفاده از پیش بینی کننده پیکسل قبلی جهت گرا در مثال  $8.19$  به دست آمده است.

### MOTION بقیه پیش بینی های جبران خسارت

همانطور که در مثال ۸.۲۰ مشاهده کردید، فریم های متوالی در یک دنباله ویدئویی اغلب بسیار مشابه هستند. کدگذاری تفاوت های آنها می تواند افزونگی زمانی را کاهش داده و فشرده سازی قابل توجهی را ایجاد کند. با این



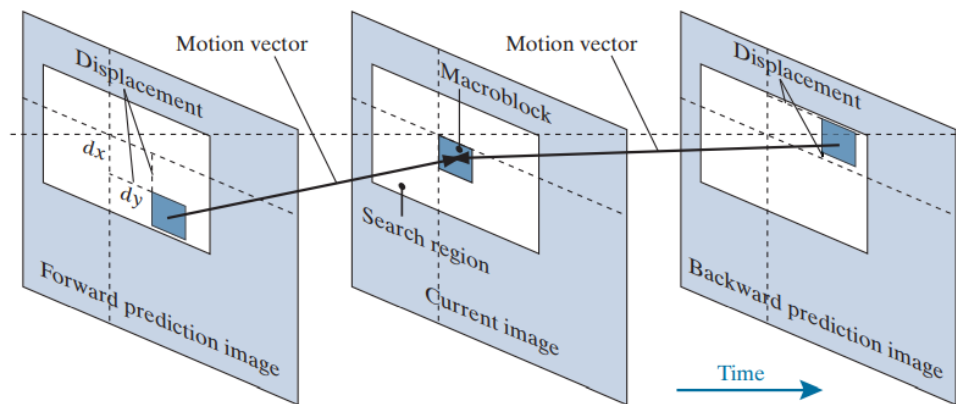
حال ، وقتی دنباله ای از فریم ها حاوی اجسام به سرعت در حال حرکت است-یا شامل بزرگنمایی و حرکت دوربین ، تبدیل ناگهانی صحنه ، یا محو شدن و محو شدن می شود-شبهات بین فریم های مجاور کاهش می یابد و فشرده سازی تحت تأثیر منفی قرار می گیرد. یعنی مانند اکثر تکنیک های فشرده سازی (مثال ۸.۵ را ببینید) ، برنامه نویسی پیش بینی مبتنی بر زمان با انواع ورودی ها ، یعنی دنباله ای از تصاویر با افزونگی زمانی قابل توجه ، بهتر کار می کند. هنگامی که در تصاویر با افزونگی زمانی کمی استفاده می شود ، ممکن است گسترش داده ها رخ دهد. سیستم های فشرده سازی ویدئو به دو طریق از مشکل گسترش داده ها جلوگیری می کند:

۱. با ردیابی حرکت اجسام و جبران آن در طول روند پیش بینی و تفاوت.
  ۲. با تبدیل روش کدگذاری متناوب در محقیقی که همبستگی بین چارچوب نامناسب (شبهات بین فریم ها) وجود داشته باشد تا کدگذاری پیش بینی کننده دارای مزیت باشد.
- اولین مورد از این موارد که جبران حرکت نامیده می شود ، موضوع باقی مانده این بخش است. با این حال ، قبل از اقدام ، باید توجه داشته باشیم که وقتی همبستگی بین چارچوب کافی برای موثر بودن کدگذاری پیش بینی وجود ندارد ، مشکل دوم معمولاً با استفاده از تبدیل دوبعدی بلوک گرا ، مانند کدگذاری مبتنی بر DCT JPEG (به بخش قبلی مراجعه کنید) برطرف می شود. به فریم هایی که به این روش فشرده می شوند (یعنی بدون پیش بینی باقیمانده) داخل قاب یا فریم مستقل (فریم I) می گویند. می توان آنها را بدون دسترسی به فریم های دیگر در ویدیویی که به آن تعلق دارند رمزگشایی کرد. فریم های I معمولاً شبیه تصاویر رمزگذاری شده با JPEG هستند و نقطه شروع ایده آلی برای تولید باقی مانده های پیش بینی هستند. علاوه بر این ، آنها درجه بالایی از دسترسی سریع ، سهولت ویرایش و مقاومت در برابر انتشار خطای تبدیل را ارائه می دهند. در نتیجه ، همه استانداردها مستلزم قرار دادن دوره ای فریم های I در جریان کد ویدیویی فشرده هستند.
- شکل ۸.۳۳ مبانی برنامه نویسی پیش بینی کننده با جبران حرکت را نشان می دهد. هر فریم ویدئویی به نواحی مستطیلی غیر همپوشانی (معمولاً در اندازه  $4 * 4$  تا  $16 * 16$ ) تقسیم می شود که ماکروبلوک نامیده می شوند. (فقط یک ماکروبلوک در شکل ۸.۳۳ نشان داده شده است.) "حرکت" هر ماکروبلوک نسبت به "محتمل ترین" موقعیت آن در قاب ویدیویی قبلی (یا بعدی) ، به نام قاب مرجع ، در بردار حرکت کدگذاری می شود. بردار با تعریف تبدیل مکان افقی و عمودی از موقعیت "به احتمال زیاد" حرکت را توصیف می کند. جابجایی ها معمولاً با دقت دقیق پیکسل ،  $1/2$  پیکسل یا  $1/4$  پیکسل مشخص می شوند. اگر از دقت پیش پیکسل استفاده می شود ، پیش بینی ها باید در هم تنیده شوند [برای مثال ، با استفاده از درون یابی دو خطی (بخش ۲.۴ را ببینید)] از ترکیبی از پیکسل ها در قاب مرجع. یک قاب رمزگذاری شده که بر اساس قاب قبلی است (یک پیش بینی رو به جلو در شکل ۸.۳۳) ، قاب پیش فرض (قاب P) نامیده می شود. قاب دیگری که بر اساس فریم بعدی است (یک

پیش بینی عقب در شکل ۸.۳۳)، قاب دو جهته (قاب B) نامیده می شود. فریم B نیاز به تبدیل مجدد جریان کد فشرده دارد تا فریم ها در ترتیب رمزگشایی مناسب، به جای ترتیب نمایش طبیعی، به رمزگشایی ارائه شوند. همانطور که ممکن است انتظار داشته باشید، برآورد حرکت جزء کلیدی برای جبران حرکت است. در حین برآورد حرکت، حرکت اجسام اندازه گیری شده و به صورت بردارهای حرکت کدگذاری می شوند. جستجوی "بهترین" بردار حرکت مستلزم تعیین معیار بهینه است. به عنوان مثال، بردارهای حرکت ممکن است بر اساس حداکثر همبستگی یا حداقل خطا بین پیکسل های ماکروبلوک و پیکسل های از پیش تعیین شده (یا پیکسل های درون یابی شده برای بردارهای حرکت زیر پیکسل) از چارچوب مرجع انتخاب شده انتخاب شوند. یکی از متداول ترین معیارهای خطا، اعوجاج مطلق متوسط (MAD) است

$$MAD(x, y) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |f(x+i, y+j) - p(x+i+dx, y+j+dy)| \quad (8-36)$$

موقعیت "به احتمال زیاد" موضعی است که اندازه گیری خطا بین کلان بلوک مرجع و ماکروبلوک رمزگذاری شده را به حداقل برساند. لازم نیست که دو بلوک بازنمایی یک شیء واحد باشند، اما باید اندازه گیری خطا را به حداقل برسانند.



شکل ۸.۳۳ مشخصات حرکت Macroblock.

جایی که  $x$  و  $y$  مختصات پیکسل سمت چپ بالای ماکروبلوک  $m * n$  هستند که کدگذاری می شوند،  $dx$  و  $dy$  جابجایی هایی از چارچوب مرجع هستند که در شکل ۸.۳۳ نشان داده شده است، و  $p$  مجموعه ای از مقادیر پیش بینی شده پیکسل های ماکروبلوک است. برای برآورد بردار حرکت زیر پیکسل،  $p$  از پیکسل ها در چارچوب مرجع درون یابی می شود. به طور معمول،  $dx$  و  $dy$  باید در یک محدوده جستجوی محدود قرار گیرند (شکل ۸.۳۳ را ببینید) در اطراف هر بلوک بزرگ.

مقادیر  $\pm 8$  تا  $\pm 64$  پیکسل متداول هستند و منطقه جستجوی افقی اغلب کمی بزرگتر از ناحیه عمودی است. یک اندازه گیری خطای محاسباتی کارآمدتر، که مجموع تعریفات مطلق (SAD) نامیده می شود، عامل  $mn$  را در معادله (۸-۳۶) حذف می کند.

با توجه به معیار انتخاب مانند معادله (۸-۳۶) ، برآورد حرکت با جستجوی  $dx$  و  $dy$  انجام می شود که MAD را در محدوده مجاز جابجایی بردار حرکت ، از جمله جابجایی های زیر پیکسل به حداقل می رساند. این فرایند اغلب تطبیق بلوک نامیده می شود. یک جستجوی جامع بهترین نتیجه ممکن را تضمین می کند ، اما از نظر محاسباتی گران است زیرا هر حرکت احتمالی باید در کل محدوده جابجایی آزمایش شود. برای بلوک های ۱۶ اینچی و محدوده جابجایی ۳۲ پیکسل (در مورد فیلم های اکشن و ورزشی دور از ذهن نیست) ، هنگام استفاده از دقت جابجایی عدد صحیح ،  $16 * 16 * 4225$  محاسبه MAD باید برای هر ماکروبلوک در یک قاب انجام شود. اگر دقت  $1/2$  یا  $1/4$  پیکسل مورد نظر است ، تعداد محاسبات به ترتیب در ضریب ۴ یا ۱۶ ضرب می شود. الگوریتم های جستجوی سریع می توانند بار محاسباتی را کاهش دهند ، اما ممکن است بردارهای حرکت بهینه را ارائه دهند یا ندهند. تعدادی الگوریتم برآورد سریع حرکت مبتنی بر بلوک در ادبیات پیشنهاد و مطالعه شده است (به Furht و همکاران [۱۹۹۷] یا Mitchell et al [۱۹۹۷] مراجعه کنید).

مثال ۸.۲۱: پیش بینی جبران شده حرکت.

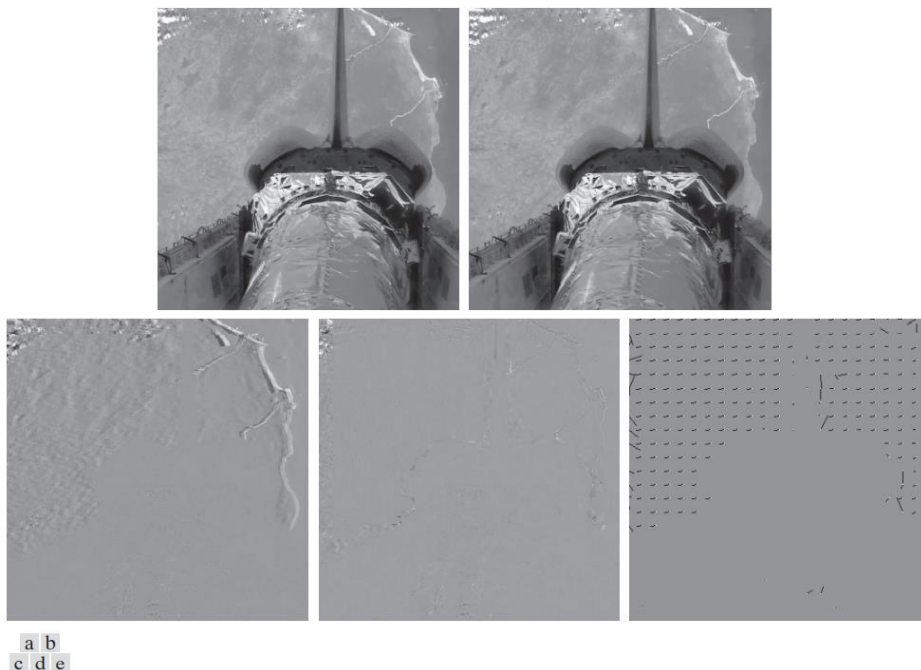
شکل ۸.۳۴ (a) و (b) از همان دنباله ویدئویی ناسا که در مثالهای ۸.۱۹ و ۸.۲۰ شکل ۸.۳۴ (ب) مشابه شکلها است. ۸.۳۱ (الف) و ۸.۳۲ (ب) ؛ شکل ۸.۳۴ (الف) بخش مربوط به یک قاب است که سیزده فریم زودتر رخ داده است. شکل ۸.۳۴ (ج) تفاوت بین دو قاب است که در محدوده شدت کامل مقیاس شده است. توجه داشته باشید که این تفاوت در ناحیه شاتل ثابت (نسبت به دوربین) ۰ است ، اما به دلیل حرکت نسبی زمین ، تفاوت های قابل توجهی در بقیه تصویر وجود دارد. انحراف استاندارد باقی مانده پیش بینی در شکل ۸.۳۴ (ج)  $12.73$  سطح شدت است. آنتروپی آن [با استفاده از معادله (۸-۷)]  $4.17$  بیت پیکسل است. حداکثر فشردن سازی قابل دستیابی ، هنگامی که طول متغیر را برای پیش بینی باقیمانده کدگذاری می کنیم ،  $1,92$  ،  $17$  ،  $8$  C است.

شکل ۸.۳۴ (d) باقیمانده پیش بینی جبران شده حرکت را با انحراف استاندارد بسیار پایین تر ( $5.62$  در مقابل سطوح شدت  $12.73$ ) و آنتروپی کمی پایین تر ( $3.04$  در مقابل  $4.17$  بیت پیکسل) نشان می دهد. آنتروپی با استفاده از معادله محاسبه شد. (۸-۷). اگر پیش بینی باقیمانده در شکل ۸.۳۴ (d) دارای طول متغیر باشد ، نسبت تراکم حاصل  $2,63 * 2,63 * 8,0$  C است. برای تولید این باقیمانده از پیش بینی ، شکل ۸.۳۴ (ب) را به  $16$  ماکروبلوک  $16 * 16$  غیر همپوشان تقسیم کردیم و هر ماکروبلوک را با هر ناحیه  $16 * 16$  در شکل ۸.۳۴ (a) -چارچوب مرجع- که در  $16 * 16$  پیکسل قرار داشت مقایسه کردیم. از موقعیت کلان بلوک در (ب). سپس از معادله استفاده کردیم. (۳۶-۳۸) برای تعیین بهترین تطابق با انتخاب جابجایی ( $dy, dx$ ) با کمترین MAD. جابجایی های حاصل اجزای  $x$  و  $y$  بردارهای حرکت هستند که در شکل ۸.۳۴ (e) نشان داده شده است. نقاط سفید در تصویر سر بردارهای حرکت هستند. آنها گوشه سمت چپ بالای ماکروبلوک های کد شده را نشان می دهند. همانطور که از الگوی بردارها مشاهده می کنید ، حرکت غالب در تصویر از چپ به راست است. در قسمت پایین تصویر ، که مربوط به مساحت شاتل فضایی در تصویر اصلی است ، هیچ حرکتی وجود ندارد و بنابراین هیچ

بردار حرکتی نمایش داده نمی شود. بلوک های کلان در این منطقه از کلان بلوک های مشابه (یعنی مربوطه) در چارچوب مرجع پیش بینی شده است. از آنجا که بردارهای حرکت در شکل ۸.۳۴ (e) بسیار همبسته هستند، می توان آنها را با طول متغیر کدگذاری کرد تا نیازهای ذخیره سازی و تبدیل آنها را کاهش دهد. تفاوت بصری بین شکل ها. ۸.۳۴ (ج) و ۸.۳۵ (الف) به دلیل مقیاس بندی است. تصویر در شکل ۸.۳۵ (a) برای مطابقت با شکل ۸.۳۵ (ب) - (د) مقیاس بندی شده است.

شکل ۸.۳۵ افزایش دقت پیش بینی را نشان می دهد که با جبران حرکت زیر پیکسل امکان پذیر است. شکل ۸.۳۵ (الف) از شکل ۸.۳۴ (ج) تکرار شده و به عنوان نقطه مرجع درج شده است. این خطای پیش بینی را نشان می دهد که بدون جبران حرکت ایجاد می شود. تصاویر در شکل ها. ۸.۳۵ (b)، (c) و (d) باقیمانده های پیش بینی جبران شده حرکت هستند. آنها بر اساس همان دو فریم هستند که در مثال ۸.۲۱ مورد استفاده قرار گرفتند و با جابجایی ماکروبلوک ها به ترتیب با وضوح ۱،  $\frac{1}{2}$  و  $\frac{1}{4}$  پیکسل (یعنی دقیق) محاسبه شدند. بلوک های ماکرو در اندازه  $8 * 8$  استفاده شد. جابجایی ها به  $8 \pm$  پیکسل محدود می شود.

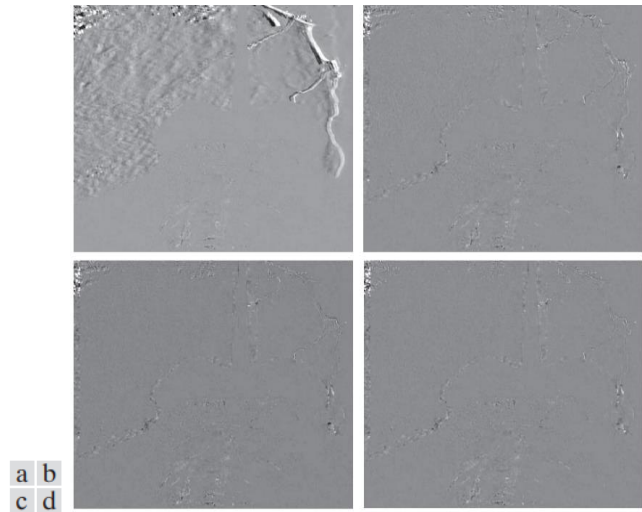
مهمترین تفاوت بصری بین باقی مانده های پیش بینی در شکل ۸.۳۵ تعداد و اندازه قله ها و دره های شدت - تاریک ترین و روشن ترین مناطق شدت آنها است. باقیمانده  $\frac{1}{4}$  پیکسل در شکل ۸.۳۵ "est (d) گواهی" از چهار تصویر است، با کمترین گردش به سیاه یا سفید. همانطور که انتظار می رفت، باریک ترین هیستوگرام را دارد. انحراف استاندارد باقی مانده های پیش بینی در شکل ها. با افزایش دقت بردار حرکت از ۱۲.۷ به ۴.۴، ۴ و ۳.۸ پیکسل به ترتیب ۸.۳۵ (الف) تا (د) کاهش می یابد. آنتروپی های باقیمانده، که با استفاده از معادله (۸-۷)، مشخص می شود. به ترتیب ۴.۱۷، ۳.۳۴، ۳.۳۵ و ۳.۳۴ بیت پیکسل هستند. بنابراین، باقیمانده های جبران شده حرکتی با وجود این که باقی مانده ها در شکل ۱ نشان داده شده، تقریباً به همان میزان اطلاعات دارند. ۸.۳۵ (c) و (d) از بیت های اضافی برای قرار دادن درون یابی  $\frac{1}{2}$  و  $\frac{1}{4}$  پیکسل استفاده می کنند. در نهایت، ما توجه داریم که یک نوار آشکار از افزایش خطای پیش بینی در سمت چپ هر حرکت باقی مانده جبران شده وجود دارد. این به دلیل حرکت چپ به راست زمین است، که مناطق جدید یا قبلاً دیده نشده از زمین را در سمت چپ هر تصویر وارد می کند. از آنجا که این مناطق در فریم های قبلی وجود ندارند، بدون در نظر گرفتن دقت مورد استفاده برای محاسبه بردارهای حرکت، نمی توان آنها را به طور دقیق پیش بینی کرد.



تصویر ۸.۳۴ (الف) و (ب) دو نمای زمین که در یک فیلم شاتل فضایی در حال گردش ، سیزده فریم از یکدیگر فاصله دارند. (ج) تصویر خطای پیش بینی بدون جبران حرکت. (د) باقیمانده پیش بینی با جبران حرکت. (ه) بردارهای حرکت مربوط به (د). نقاط سفید در (e) نشان دهنده سر پیکان بردارهای حرکتی است که به تصویر کشیده شده است. (تصاویر اصلی از ناسا گرفته شده است).

برآورد حرکت یک کار محاسباتی سخت است. خوشبختانه فقط رمزگذار باید حرکت ماکروبلوک را تخمین بزند. با توجه به بردارهای حرکتی کلان بلوک ها ، رمزگشایی به سادگی به مناطق فریم های مرجع دسترسی پیدا می کند که در رمزگذار برای ایجاد باقی مانده های پیش بینی استفاده می شود.

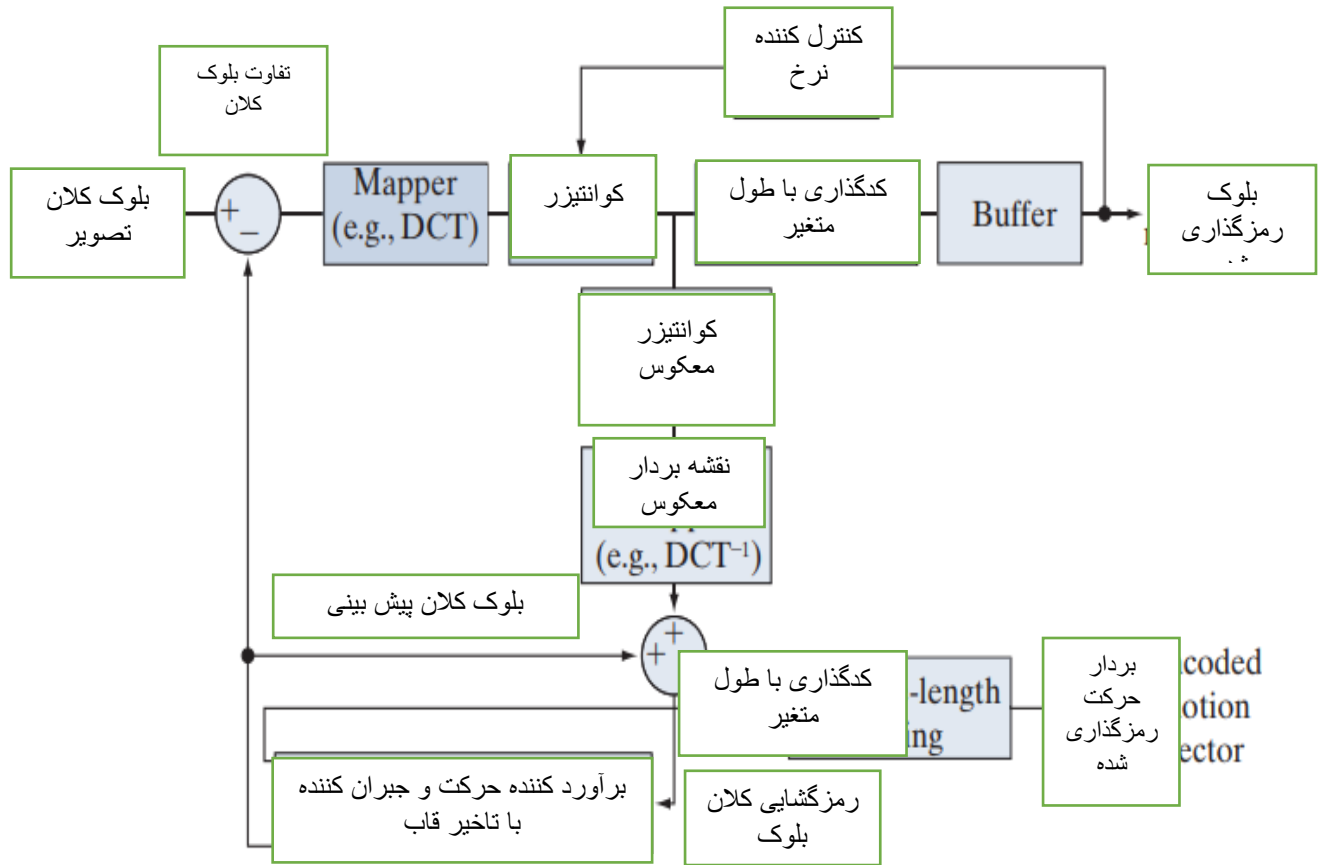
به همین دلیل ، برآورد حرکت در اکثر استانداردهای فشرده سازی ویدئو گنجانده نشده است. استانداردهای فشرده سازی بر رمزگشایی تمرکز می کنند ، محدودیت هایی را بر ابعاد کلان بلوک ، پیش بینی بردار حرکت ، محدوده جابجایی افقی و عمودی و موارد مشابه قرار می دهند. جدول ۸.۱۲ پارامترهای کدگذاری پیش بینی کننده کلیدی برخی از مهمترین استانداردهای فشرده سازی ویدئو را ارائه می دهد. توجه داشته باشید که در اکثر استانداردها از  $8 * 8$  DCT برای رمزگذاری فریم ۱ استفاده می شود ، اما برای جبران حرکت ، مساحت بزرگتری (به عنوان مثال ،  $16 * 16$  ماکروبلوک) مشخص شده است. علاوه بر این ، حتی باقی مانده های پیش بینی قاب P و B نیز به دلیل کارایی کمی سازی ضریب DCT ، کدگذاری می شوند. در نهایت ، ما توجه داریم که استانداردهای H.۲۶۴ و MPEG-۴ AVC از کدگذاری پیش بینی درون چارچوب (در فریم های ۱) برای کاهش افزونگی فضایی پشتیبانی می کنند.



شکل ۸.۳۵ حرکت پیش پیکسل باقیمانده های پیش بینی را جبران می کند: (الف) بدون جبران حرکت. (ب) دقت یک پیکسل ؛ (ج) دقت پیکسل ؛ و (د) دقت پیکسل. (تمام خطاهای پیش بینی در محدوده شدت کامل مقیاس بندی شده و سپس در ۲ ضرب می شوند تا دید آنها افزایش یابد.)  
جدول ۸.۱۲ کدگذاری پیش بینی کننده در استانداردهای فشرده سازی ویدئو.

پارامتر	H.261	MPEG-1	H.262 MPEG-2	H.263	MPEG-4	VC-1 WMV-9	H.264 MPEG-4 AVC
دقت بردار حرکت	1	1/2	1/2	1/2	1/4	1/4	1/4
اندازه های ماکرو بلاک	16 × 16	16 × 16	16 × 16 16 × 8	16 × 16 8 × 8	16 × 16 8 × 8	16 × 16 8 × 8	16 × 16 16 × 8 8 × 8 8 × 4 4 × 8 4 × 4
تبدیل	8 × 8 DCT	8 × 8 DCT	8 × 8 DCT	8 × 8 DCT	8 × 8 DCT	8 × 8 8 × 4 4 × 8 4 × 4 Integer DCT	4 × 4 8 × 8 Integer
پیش بینی های بین فریم	P	P, B	P, B	P, B	P, B	P, B	P, B
درون قاب پیش بینی ها	No	No	No	No	No	No	Yes

تعیین مقدار همانطور که در قسمت قبل تعریف شده است برگشت ناپذیر است. "کوانتیزر معکوس" در فصل ۸.۱۷ مانع از دست رفتن اطلاعات نمی شود.



شکل ۸.۳۶ یک رمزگذار ویدیویی با حرکت معمولی جبران می کند

مثال ۸.۲۲: مثال فشرده سازی ویدئو.

ما بحث خود را در مورد برنامه نویسی پیش بینی جبران شده با یک مثال با نشان دادن نوع فشرده سازی که با روشهای فشرده سازی ویدئویی مدرن امکان پذیر است ، به پایان می بریم. شکل ۸.۳۷ fi پانزده فریم ویدئویی ۱ دقیقه ای (۷۲۰ \* ۱۲۸۰) HD تمام رنگی ناسا را نشان می دهد که قسمتهایی از آن در سراسر این بخش استفاده شده است. اگرچه تصاویر نشان داده شده تک رنگ هستند ، ویدئو دنباله ای از ۱۸۲۹ فریم تمام رنگ است. توجه داشته باشید که صحنه های متنوعی ، حجم زیادی از حرکت و جلوه های محو شدن متعدد وجود دارد. به عنوان مثال ، ویدئو با ۱۵۰ فریم محو از رنگ مشکی ، که شامل فریم های ۲۱ و ۴۴ در شکل ۸.۳۷ است ، باز می شود و با دنباله ای محو شامل فریم های ۱۵۹۵ ، ۱۶۰۹ و ۱۶۵۲ در شکل ۸.۳۷ ، به پایان می رسد. در نهایت محو شدن به رنگ سیاه چندین تبدیل ناگهانی صحنه نیز وجود دارد ، مانند تبدیل در فریم های ۱۳۰۳ و ۱۳۰۴ در شکل ۸.۳۷. نسخه فشرده H.۲۶۴ ویدئوی ناسا که به صورت Quicktime File ذخیره می شود (جدول ۸.۵ را ببینید) نیاز دارد

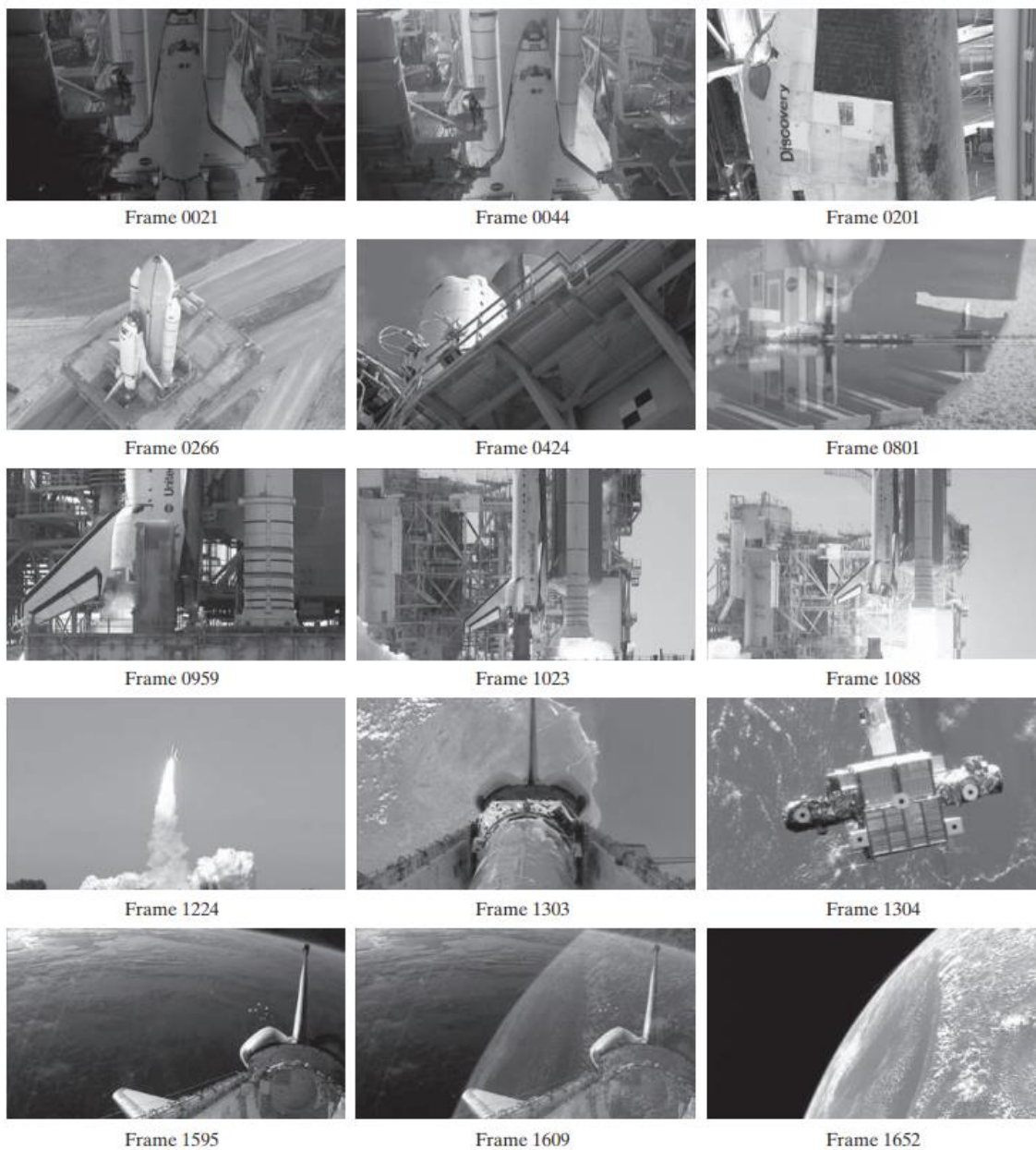
۴۴.۵۶ مگابایت فضای ذخیره سازی ، علاوه ۱.۳۹ مگابایت دیگر برای صدای مرتبط. کیفیت فیلم عالی است. حدود ۵ گیگابایت داده برای ذخیره فریم های ویدئویی به عنوان تصاویر فشرده تمام رنگ نیاز است. لازم به ذکر

است که ویدئو شامل توالی شامل چرخش و تبدیل مقیاس است (به عنوان مثال ، دنباله شامل فریم های ۹۵۹ ، ۱۰۲۳ و ۱۰۸۸ در شکل ۸.۳۷). اما بحث در این بخش تنها به تبدیل محدود شده است. (برای بخش ویدیویی ناسا که در این مثال استفاده شده است به وب سایت کتاب مراجعه کنید).

### کدگذاری پیش بینی LOSSY

در این بخش ، ما یک کوانتیزر به مدل کدگذاری پیش بینی بدون ضرر که قبلاً معرفی شده بود ، اضافه می کنیم و جابجایی بین دقت بازسازی و تابع فشرده سازی را در زمینه پیش بینی کننده های فضایی بررسی می کنیم. همانطور که در شکل ۸.۳۸ نشان داده شده است ، کوانتیزر ، که جایگزین نزدیکترین تابع صحیح رمزگذار بدون خطا است ، بین رمزگذار نماد و نقطه ای که خطای پیش بینی در آن شکل می گیرد ، وارد می شود. این خطای پیش بینی را در محدوده محدودی از خروجی ها ، با  $e^-(n)$  ، که مقدار فشرده سازی و اعوجاج رخ می دهد ، ترسیم می کند.





شکل ۸.۳۷ پانزده فریم از یک ویدیوی ۱۸۲۹ فریم ۱۸۲۹ ناسا. ویدئوی اصلی به صورت تمام رنگی HD است. (با اجازه ناسا.)

به منظور جایگذاری مرحله کوانتیزاسیون ، رمزگذار بدون خطا در شکل ۸.۳۰ (a) باید تبدیل کند تا پیش بینی های ایجاد شده توسط رمزگذار و رمزگشایی معادل باشند. همانطور که شکل ۸.۳۸ (a) نشان می دهد ، این امر با قرار دادن پیش بینی کننده رمزگذار تلفات در یک حلقه بازخورد ، جایی که ورودی آن به صورت  $f(n)$  ، به عنوان تابعی از پیش بینی های گذشته و مقاردهی مربوطه تولید می شود خطاها به این معنا که،

$$\hat{f}(n) = e(n) + \hat{f}(n) \quad (8-37)$$

جایی که  $\hat{f}(n)$  همانطور که قبلاً تعریف شده است این پیکربندی حلقه بسته از خطا جلوگیری می کند ایجاد خروجی رمزگشایی در شکل ۸.۳۸ (ب) توجه داشته باشید که خروجی رمزگشایی نیز با معادله (۸-۳۷) نشان داده شده است.

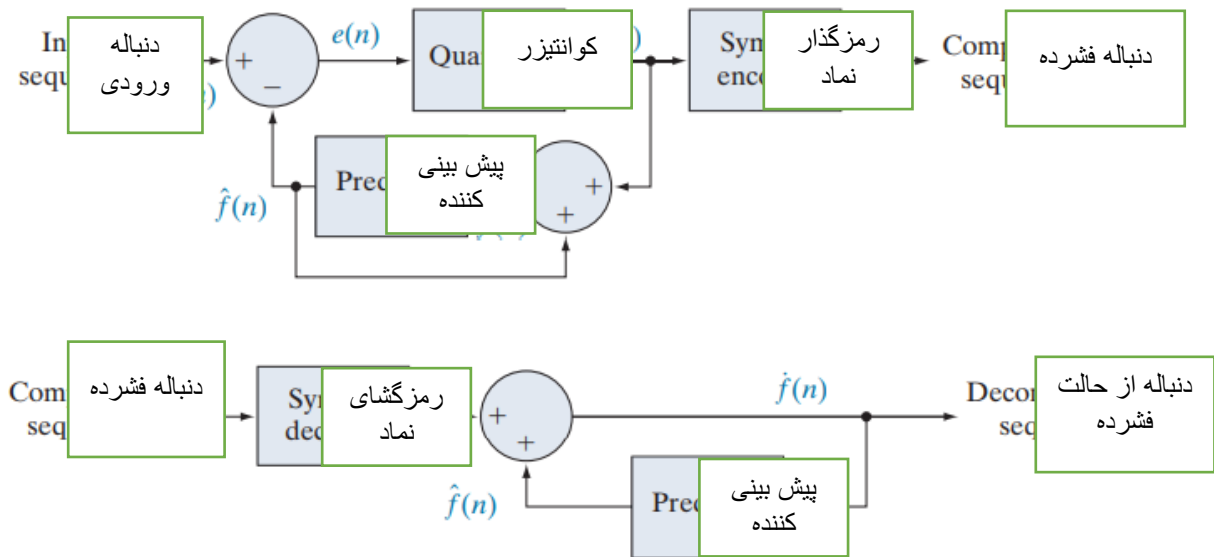
مثال ۸.۲۳: مدولاسیون دلتا.

مدولاسیون دلتا (DM) یک شکل ساده اما شناخته شده از کدگذاری پیش بینی کننده زیان آور است که در آن پیش بینی کننده و کوانتیزه به عنوان

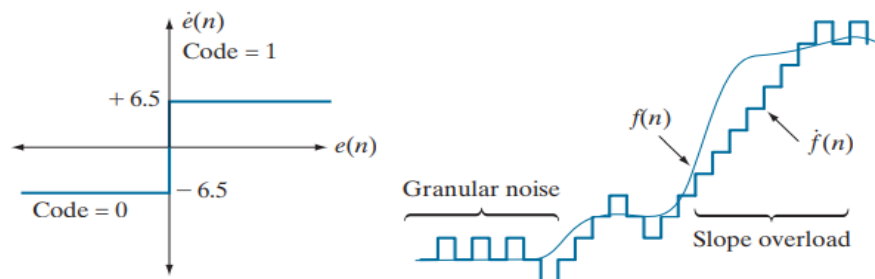
$$\hat{f}(n) = \alpha \hat{f}(n-1) \quad (8-38)$$

$$e(n) = \begin{cases} +\zeta & \text{for } e(n) > 0 \\ -\zeta & \text{otherwise} \end{cases} \quad (8-39)$$

جایی که  $\hat{f}(n)$  همانطور که قبلاً تعریف شده است این پیکربندی حلقه بسته از خطا جلوگیری می کند ایجاد خروجی رمزگشایی در شکل ۸.۳۸ (ب) توجه داشته باشید که خروجی رمزگشایی نیز با معادله (۸-۳۷) نشان داده شده است.



شکل ۸.۳۸ یک مدل کدگذاری پیش بینی کننده زیان آور: الف) رمزگذار؛ ب) رمزگشایی



Input		Encoder			Decoder		Error	
$n$	$f(n)$	$\hat{f}(n)$	$e(n)$	$\hat{e}(n)$	$\dot{f}(n)$	$\hat{f}(n)$	$\dot{f}(n)$	$f(n) - \hat{f}(n)$
0	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	-5.5
2	14	20.5	-6.5	-6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	-5.5
.	.	.	.	.	.	.	.	.
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.6	53.0	59.5	17.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

شکل ۸.۳۹ نمونه ای از مدولاسیون دلتا تبدیل بدون خطا اولین نمونه ورودی به رمزگشا.

ارزیابی معادلات (۸-۳۸) ، (۸-۲۹) ، (۸-۳۹) و (۸-۳۷). بنابراین ، هنگامی که  $n = 1$  ، برای مثال ،  $\hat{f}(1) = (1)(14) = 14$  ،  $e(1) = 15 - 14 = 1$  ،  $\hat{e}(1) = +6.5$  ، زیرا  $\hat{f}(1) = 14$  ،  $f(1) > 0$  ، و

بازسازی حاصله خطا (۱۵-، ۲۰، ۵) یا -۵.۵ است.

شکل ۸.۳۹ (ب) به صورت گرافیکی داده های جدول بندی شده در شکل ۸.۳۹ (ج) را نشان می دهد. هم ورودی و هم کاملاً خروجی رمزگشایی شده  $f(n)$  و  $\hat{f}(n)$  نشان داده شده است. توجه داشته باشید که در منطقه ای که به سرعت در حال تبدیل است از  $n = 14$  به  $n = 19$  ، جایی که  $Z$  بسیار کوچک بود تا بزرگترین تبدیلات ورودی را نشان دهد ، یک اعوجاج معروف به اضافه بار شیب رخ می دهد. علاوه بر این ، هنگامی که  $Z$  برای نمایش کوچکترین تبدیلات ورودی بسیار بزرگ بود ، مانند ناحیه نسبتاً صاف از  $n = 0$  تا  $n = 7$  ، نویز دانه ای ظاهر می شود. در تصاویر ، این دو پدیده منجر به تار شدن لبه های اجسام و سطوح دانه دار یا پر نویز (یعنی مناطق صاف مخدوش) می شود.

تحریفات ذکر شده در مثال قبل در همه اشکال کدگذاری پیش بینی کننده زیان آور مشترک است. شدت این تحریفها به مجموعه مختلط ای از فعل و انفعالات بین روشهای پیش بینی و پیش بینی بستگی دارد. با وجود این فعل و انفعالات ، پیش بینی کننده به طور معمول با فرض عدم وجود خطای کوانتاسیون و کوانتیزر به منظور به

حداقل رساندن خطای خود طراحی شده است. یعنی پیش بینی کننده و کوانتیزر مستقل از یکدیگر طراحی شده اند.

### پیش بینی کننده های بهینه

در بسیاری از برنامه های کدگذاری پیش بینی کننده ، پیش بینی کننده برای به حداقل رساندن خطای پیش بینی میانگین مربع رمزگذار انتخاب می شود

$$E\{e^2(n)\} = E\{[f(n) - \hat{f}(n)]^2\} \quad (8-40)$$

علامت  $E\{\bullet\}$  نشان دهنده آمار است

مشروط به محدودیتی که

$$\hat{f}(n) = \hat{e}(n) + \hat{f}(n) \approx e(n) + \hat{f}(n) = f(n) \quad (8-41)$$

$$\hat{f}(n) = \sum_{i=1}^m \alpha_i f(n-i) \quad (8-42)$$

بدین معنا که معیار بهینه سازی حداقل خطای پیش بینی میانگین مربع است ، خطای کوانتیزاسیون ناچیز  $[\hat{e}(n) \approx e(n)]$  فرض می شود و پیش بینی به ترکیب خطی  $m$  نمونه های قبلی محدود می شود. این محدودیت ها هستند. ضروری نیست ، اما آنها تجزیه و تحلیل را به میزان قابل توجهی ساده کرده و در عین حال پیچیدگی محاسباتی پیش بینی کننده را کاهش می دهند. روش کدگذاری پیش بینی شده به عنوان مدولاسیون کد پالس افتراقی (DPCM) نامیده می شود. در این شرایط ، مشکل طراحی بهینه پیش بینی به تمرین نسبتاً ساده انتخاب معیارهای پیش بینی  $m$  که بیان را به حداقل می رساند ، کاهش می یابد.

به طور کلی ، پیش بینی کننده بهینه برای یک دنباله غیر گاوسی یک تابع غیر خطی از نمونه هایی است که برای تشکیل برآورد استفاده می شود.

$$E\{e^2(n)\} = E\left\{\left[f(n) - \sum_{i=1}^m \alpha_i f(n-i)\right]^2\right\} \quad (8-43)$$

تمایز معادله (۸-۴۳) با توجه به هر ضریب ، معادل مشتقات به صفر ، و حل مجموعه معادلات همزمان حاصل از این فرض که  $f(n)$  دارای میانگین صفر و واریانس  $S^2$  است

$$\alpha = \mathbf{R}^{-1} \mathbf{r} \quad (8-44)$$

جایی که  $\mathbf{R}^{-1}$  معکوس ماتریس همبستگی  $m \times m$  است

$$\mathbf{R} = \begin{bmatrix} E\{f(n-1)f(n-1)\} & E\{f(n-1)f(n-2)\} & \cdots & E\{f(n-1)f(n-m)\} \\ E\{f(n-2)f(n-1)\} & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ E\{f(n-m)f(n-1)\} & E\{f(n-m)f(n-2)\} & \cdots & E\{f(n-m)f(n-m)\} \end{bmatrix} \quad (8-45)$$

و  $\mathbf{r}$  و  $\mathbf{A}$  بردارهای عنصر  $m$  هستند

$$\mathbf{r} = \begin{bmatrix} E\{f(n)f(n-1)\} \\ E\{f(n)f(n-2)\} \\ \vdots \\ E\{f(n)f(n-m)\} \end{bmatrix} \quad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} \quad (8-46)$$

بنابراین برای هر دنباله ورودی، ضرایبی که معادله (۸-۴۳) را به حداقل می‌رسانند، را می‌توان از طریق یک سری عملیات ماتریسی ابتدایی تعیین کرد. علاوه بر این، ضرایب فقط به همبستگی خودکار نمونه‌ها در دنباله اصلی بستگی دارد. واریانس خطای پیش‌بینی ناشی از استفاده از این ضرایب بهینه است

$$\sigma_e^2 = \sigma^2 - \boldsymbol{\alpha}^T \mathbf{r} = \sigma^2 - \sum_{i=1}^m E\{f(n)f(n-i)\} \alpha_i \quad (8-47)$$

اگرچه مکانیک‌های ارزیابی معادله (۸-۴۴) بسیار ساده هستند، محاسبه همبستگی خودکار مورد نیاز برای تشکیل  $\mathbf{R}$  و  $\mathbf{r}$  در عمل بسیار دشوار است به طوری که پیش‌بینی‌های محلی (آنهايي که ضرایب پیش‌بینی برای هر دنباله ورودی محاسبه می‌شود) تقریباً هرگز استفاده نمی‌شود. در بیشتر موارد، مجموعه‌ای از ضرایب جهانی با فرض یک مدل ورودی ساده و جایگزینی همبستگی خودکار مربوطه به معادلات (۸-۴۵) و (۸-۴۶) بدست می‌آید. به عنوان مثال، هنگامی که یک منبع شکل دو بعدی مارکوف (بخش ۸.۱ را ببینید) با تابع خود همبستگی جداگانه

$$E\{f(x,y)f(x-i,y-j)\} = \sigma^2 \rho_v^i \rho_h^j \quad (8-48)$$

و پیش‌بینی‌کننده خطی تعمیم یافته مرتبه چهارم

$$\hat{f}(x,y) = \alpha_1 f(x,y-1) + \alpha_2 f(x-1,y-1) + \alpha_3 f(x-1,y) + \alpha_4 f(x-1,y+1) \quad (8-49)$$

فرض می‌شود، ضرایب بهینه حاصله (جین [۱۹۸۹]) عبارتند از

$$\alpha_1 = \rho_h \quad \alpha_2 = -\rho_v \rho_h \quad \alpha_3 = \rho_v \quad \alpha_4 = 0 \quad (8-50)$$

جایی که  $\rho_h$  و  $\rho_v$  به ترتیب ضرایب همبستگی افقی و عمودی شکل مورد بررسی هستند. در نهایت، مجموع ضرایب پیش‌بینی در معادله (۸-۴۲) معمولاً لازم است کمتر یا مساوی یک باشد. به این معنا که،

$$\sum_{i=1}^m \alpha_i \leq 1 \quad (8-51)$$

این محدودیت برای اطمینان از خروجی پیش بینی کننده در محدوده مجاز ورودی و کاهش تاثیر نویز تبدیل [که به طور کلی به عنوان رگه های افقی در تصاویر بازسازی شده هنگام ورودی به شکل ۸.۳۸ (a) دیده می شود] انجام می شود. یک شکل است [کاهش حساسیت رمزگشایی DPCM به نویز ورودی مهم است، زیرا یک خطای واحد (در شرایط مناسب) می تواند به همه خروجی های آینده منتقل شود. یعنی ممکن است خروجی رمزگشا ناپایدار باشد. محدودیت بیشتر معادله (۸-۵۱) به شدت کمتر از ۱، تأثیر خطای ورودی را به تعداد کمی خروجی محدود می کند.

مثال ۸.۲۴: مقایسه تکنیک های پیش بینی.

خطای پیش بینی ناشی از کدگذاری شکل تک رنگ شکل ۸.۹ (الف) را با فرض خطای کوانتیزه صفر و با هر یک از چهار پیش بینی کننده در نظر بگیرید:

$$\hat{f}(x, y) = 0.97f(x, y - 1) \quad (8-52)$$

$$\hat{f}(x, y) = 0.5f(x, y - 1) + 0.5f(x - 1, y) \quad (8-53)$$

$$\hat{f}(x, y) = 0.75f(x, y - 1) + 0.75f(x - 1, y) - 0.5f(x - 1, y - 1) \quad (8-54)$$

$$\hat{f}(x, y) = \begin{cases} 0.97f(x, y - 1) & \text{if } \Delta h \leq \Delta v \\ 0.97f(x - 1, y) & \text{otherwise} \end{cases} \quad (8-55)$$

جایی که  $\Delta h = |f(x - 1, y) - f(x - 1, y - 1)|$  و  $\Delta v = |f(x, y - 1) - f(x - 1, y - 1)|$  نشان دهنده افقی و شیب عمودی در نقطه  $(x, y)$ . معادلات (۸-۵۲) تا (۸-۵۵) مجموعه ای نسبتاً قوی از ai را ارائه می دهد که تابع رضایت بخشی را در طیف وسیعی از تصاویر ارائه می دهد. پیش بینی کننده تطبیقی معادله (۸-۵۵) به منظور بهبود بازسازی لبه ها با محاسبه اندازه گیری محلی خواص جهت شکل  $(\Delta h)$  و  $(\Delta v)$  و انتخاب پیش بینی کننده ای که مخصوص رفتار اندازه گیری شده است، طراحی شده است. شکل ۸.۴۰ (الف) تا (د) تصاویر خطای پیش بینی را که ناشی از استفاده از پیش بینی کننده ها است نشان می دهد. معادلات (۸-۵۲) تا (۸-۵۵). توجه داشته باشید که خطای محسوس بصری با افزایش ترتیب پیش بینی کاهش می یابد.  $devi$  انحرافات استاندارد خطاهای پیش بینی از الگویی مشابه پیروی می کند<sup>۴۱</sup>. شدت آنها به ترتیب ۱۱.۱، ۹.۸، ۹.۱ و ۹.۷ است.

### کوانتیزاسیون بهینه

تابع کمی سازی راه پله  $t = q(s)$  در شکل ۸.۴۱ یک تابع فرد از  $s$  [یعنی  $q(-s) = -q(s)$ ] است که می تواند به طور کامل با مقادیر  $L_2$   $Si$  و  $ti$  توصیف شود. در اولین ربع نمودار نشان داده شده است. این نقاط شکست، ناپیوستگی های تابع را تعریف می کند و به آنها سطح تصمیم گیری و بازسازی کوانتیزر می گویند. به عنوان یک امر متعامد،  $s$  در نظر گرفته می شود که در  $ti$  در صورتی که در بازه نیمه باز قرار دارد  $(si - 1, si)$  قرار گرفته

<sup>۴۱</sup> پیش بینی کننده هایی که از بیش از سه یا چهار پیکسل قبلی استفاده می کنند، فشرده سازی کمی را برای پیچیدگی بیش ساز افزایش می دهند (حبیبی [۱۹۷۱]).

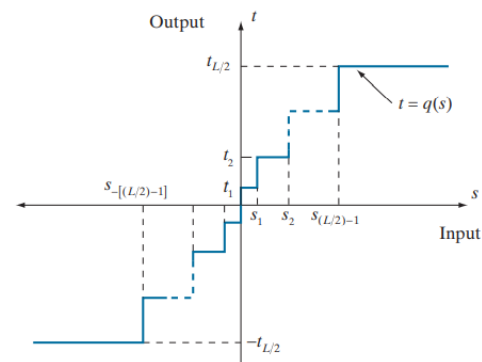
است. مشکل طراحی کوانتیزر این است که بهترین  $s_i$  و  $t_i$  را برای یک گزینه خاص انتخاب کنید. معیار اندازه گیری و تابع چگالی احتمال ورودی  $p(s)$  اگر بهینه سازی معیار، که می تواند یک معیار آماری یا روانی-بصری باشد،  $†$  به حداقل رساندن خطای کوانتیزاسیون میانگین مربع است که  $FB(s_i, t_i)$  و  $p(s)$  یک تابع زوج است، شرایط برای حداقل خطا (حداکثر [۱۹۶۰]) هستند



شکل ۸.۴۰ مقایسه چهار روش پیش بینی خطی

$$\int_{s_{i-1}}^{s_i} (s - t_i) p(s) ds = 0 \quad i = 1, 2, \dots, \frac{L}{2} \quad (8-56)$$

$$s_i = \begin{cases} 0 & i = 0 \\ \frac{t_i + t_{i+1}}{2} & i = 1, 2, \dots, \frac{L}{2} - 1 \\ \infty & i = \frac{L}{2} \end{cases} \quad (8-57)$$



شکل ۸.۴۱ یک تابع کوانتیزه معمولی

<sup>۴۲</sup> برای اطلاعات بیشتر در مورد اقدامات روانی و بصری به [۱۹۷۷] Netravali و Limb مراجعه کنید.

$$s_{-i} = -s_i \quad t_{-i} = -t_i \quad (8-58)$$

معادله (۵۶-۸) نشان می دهد که سطوح بازسازی سانتروئیدهای مناطق تحت  $p(s)$  در فواصل تصمیم گیری تعیین شده هستند ، در حالی که معادله (۵۷-۸) نشان می دهد که سطح تصمیم گیری در نیمه راه بین سطوح بازسازی قرار دارد.

معادله (۵۸-۸) نتیجه این واقعیت است که  $q$  یک تابع فرد است. برای هر  $L$ ،  $s_i$  و  $t_i$  که معادله (۵۶-۸) تا (۸-۸) را برآورده می کنند. از نظر خطای میانگین مربع مطلوب هستند. کوانتیزر مربوطه یک کوانتیزر Lloyd-Max سطح  $L$  نامیده می شود. جدول ۸.۱۳ سطوح تصمیم گیری و بازسازی Lloyd-Max در سطح ۲، ۴ و ۸ را برای یک واریانس واحد تابع چگالی احتمال لاپلاسی لیست می کند [به معادله مراجعه کنید. (۸-۳۴)]. زیرا به دست آوردن راه حل صریح یا بسته برای معادلات (۵۶-۸) تا (۵۸-۸) برای اکثر  $p$  (های) غیرحادی مشکل است ، این مقادیر به صورت عددی تولید شده اند (Paez and Glisson [۱۹۷۲]). سه کوانتیزر نشان داده شده به ترتیب نرخ خروجی ثابت ۱، ۲ و ۳ بیت پیکسل را ارائه می دهند. همانطور که جدول ۸.۱۳ برای توزیع واریانس واحد ساخته شده است ، سطح بازسازی و تصمیم گیری در مورد  $S-1$  با ضرب مقادیر جدول بندی شده در انحراف استاندارد چگالی احتمال بدست می آید.

جدول ۸.۱۳ کوانتیزرهای Lloyd-Max برای تابع چگالی احتمال لاپلاسی و واحد واریانس.

Levels	2		4		8	
	$s_i$	$t_i$	$s_i$	$t_i$	$s_i$	$t_i$
1	$\infty$	0.707	1.102	0.395	0.504	0.222
2			$\infty$	1.810	1.181	0.785
3					2.285	1.576
4					$\infty$	2.994
$\theta$	1.414		1.087		0.731	

تابع مورد بررسی ردیف نهایی جدول اندازه گام ،  $u$  را نشان می دهد که معادله های (۵۶-۸) تا (۵۸-۸) و همزمان را برآورده می کند. محدودیت اضافی که

$$t_i - t_{i-1} = s_i - s_{i-1} = \theta \quad (8-59)$$

اگر از رمزگذار نماد که از کد طول متغیر استفاده می کند در رمزگذار پیشگویی کننده زیان آور شکل ۸.۳۸ (a) استفاده شود ، یک کوانتیزر یکنواخت بهینه با اندازه گام  $u$  نرخ کد پایین تری (برای PDF لاپلاسی) نسبت به یک ثابت ارائه می دهد. طول کد Lloyd - حداکثر کوانتیزر با همان وفاداری خروجی (O'Neil [۱۹۷۱]). اگرچه Lloyd-Max و کوانتیزرهای بهینه یکنواخت سازگار نیستند ، اما با تنظیم سطوح کوانتیزاسیون بر اساس رفتار محلی یک شکل می توان به بهره زیادی دست یافت. از نظر تئوری ، مناطق به آرامی در حال تبدیل را



می توان به طور کامل کوانتیزه کرد ، در حالی که مناطق در حال تبدیل سریع درشت تر هستند. این روش همزمان نویز دانه ای و اضافه بار شیب را کاهش می دهد ، در حالی که تنها به افزایش حداقل میزان کد نیاز دارد. این تجارت پیچیدگی کوانتیزر را افزایش می دهد.

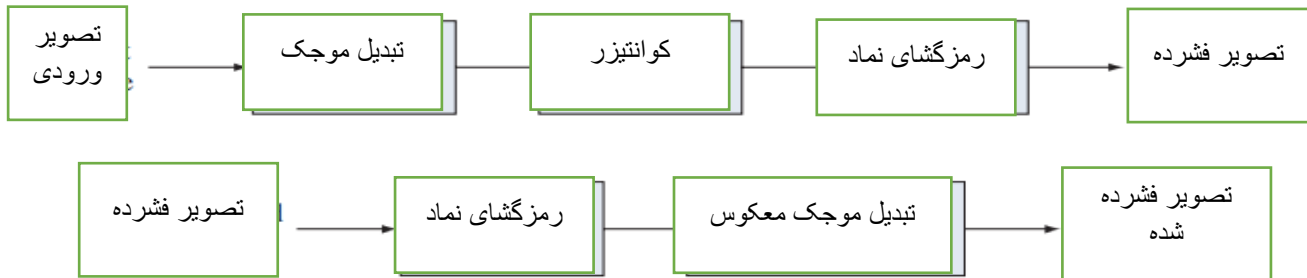
اگر از رمزگذار نماد که از کد طول متغیر استفاده می کند در رمزگذار پیشگویی کننده زیان آور شکل ۸.۳۸ (a) استفاده شود ، یک کوانتیزر یکنواخت بهینه با اندازه گام  $u$  نرخ کد پایین تری (برای PDF لاپلاسین ) نسبت به یک ثابت ارائه می دهد. طول کد Lloyd - حداکثر کوانتیزر با همان وفاداری خروجی (O'Neil [۱۹۷۱]).

با توجه به جداول ۸.۳-۸.۵ ، از کدگذاری موجک در • JPEG-۲۰۰۰ استاندارد فشرده سازی

## ۸.۱۱ کدگذاری موجک

همانطور که در مورد تکنیک های کدگذاری تبدیل بلوکی که قبلاً ارائه شد ، کدگذاری موجک بر این ایده استوار است که ضرایب دگرگونی که پیکسل های یک شکل را از هم جدا می کند ، می تواند کارآمدتر از خود پیکسل های اصلی کدگذاری شود. اگر تابع های اصلی تبدیل (در این مورد موجک) بیشتر اطلاعات بصری مهم را در تعداد کمی از ضرایب قرار دهد ، ضرایب باقی مانده را می توان به صورت درشت اندازه گیری کرد یا با اعوجاج کوچک شکل به صفر رساند. شکل ۸.۴۲ یک سیستم کدگذاری موجک معمولی را نشان می دهد. برای کدگذاری یک شکل  $2^l \times 2^l$  ، یک موجک تجزیه و تحلیل ،  $C$  و حداقل سطح تجزیه ،  $P - J$  ، انتخاب شده و برای محاسبه تبدیل موجک گسسته شکل استفاده می شود. اگر موجک دارای مقیاس بندی  $W W$  باشد ، می توان از تبدیل موجک سریع (بخش ۷.۱۰ را ببینید) استفاده کرد. در هر صورت ، تبدیل محاسبه شده بخش بزرگی از شکل اصلی را به ضرایب تجزیه افقی ، عمودی و مورب با میانگین صفر و احتمالات شبیه به لاپلاسی تبدیل می کند. از آنجا که بسیاری از همفکران محاسبه شده اطلاعات بصری کمی دارند ، می توان آنها را کوانتیزه و کدگذاری کرد تا افزونگی ضریب ضریب و کدگذاری را به حداقل برساند. علاوه بر این ، کمی سازی می تواند برای بهره برداری از همبستگی موقعیتی در سطوح تجزیه  $P$  مناسب باشد. یک یا چند روش کدگذاری بدون ضرر ، مانند طول اجرا ، هافمن ، حساب و کدگذاری صفحه بیت ، می تواند در مرحله کدگذاری نماد نهایی گنجانده شود. رمزگشایی با وارونه کردن عملیات رمزگذاری ، به استثنای کوانتیزاسیون ، انجام می شود که نمی توان دقیقاً معکوس کرد.

تفاوت اصلی بین سیستم موجک شکل ۸.۴۲ و سیستم کدگذاری تبدیل شکل ۸.۲۱ حذف مراحل پردازش زیر شکل کدگذار تبدیل است. از آنجایی که تبدیل موجک از نظر محاسباتی موثر است و ذاتاً محلی است (یعنی تابع های اصلی آنها در مدت زمان محدود است) ، تقسیم بندی شکل اصلی غیر ضروری است. همانطور که بعداً در این بخش خواهید دید ، حذف مرحله تقسیم ، مصنوع مسدود کننده ای را که تقریب های مبتنی بر DCT را در نسبت های فشرده سازی بالا مشخص می کند ، حذف می کند.



شکل ۸.۴۲ سیستم کدگذاری موجک: الف) رمزگذار؛ ب) رمزگشایی انتخاب موج

موجک هایی که به عنوان پایه تبدیلات رو به جلو و معکوس در شکل ۸.۴۲ انتخاب شده اند، بر همه جنبه های طراحی و تابع سیستم کدگذاری موجک تأثیر می گذارد. آنها به طور مستقیم بر پیچیدگی محاسباتی تبدیلات و، به طور مستقیم، بر توانایی سیستم برای فشرده سازی و بازسازی تصاویر خطای قابل قبول تأثیر می گذارند. هنگامی که موجک تبدیل کننده دارای یک مقیاس دهی همراه است، تبدیل می تواند به عنوان دنباله ای از عملیات فیلترینگ دیجیتال، با تعداد ضربه های فیلتر برابر با تعداد موجک های غیر صفر و ضرایب بردار مقیاس بندی پیاده سازی شود. قابلیت موجک برای بسته بندی اطلاعات در تعداد کمی از ضرایب تبدیل، فشرده سازی و بازسازی تابع آن را تعیین می کند.

موجک های Daubechies و موجک های دو وجهی بیشترین کاربرد گسترده ای در توابع انبساط برای فشرده سازی مبتنی بر موجک دارند. دومی اجازه می دهد تا خواص تجزیه و تحلیل مفید، مانند تعداد لحظه های ناپدید شدن (به بخش ۷.۱۰ مراجعه کنید)، در فیلترهای تجزیه وارد شوند، در حالی که ویژگیهای مهم ترکیب، مانند صاف شدن بازسازی، در فیلترهای بازسازی ساخته شده اند.

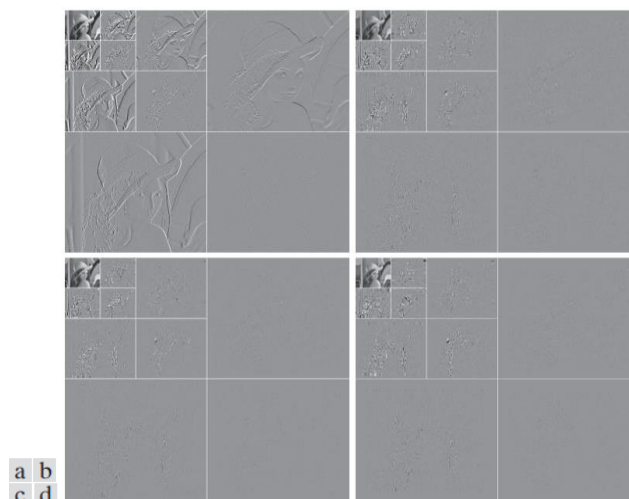
مثال ۸.۲۵: پایه های موجک در کدگذاری موجک.

شکل ۸.۴۳ شامل چهار تبدیل موجک گسسته از شکل ۸.۹ (الف) است. موجک های هار، ساده ترین و تنها موجک های ناپیوسته که در این مثال در نظر گرفته شده اند، به عنوان توابع انبساط یا پایه در شکل ۸.۴۳ (a) استفاده شده اند. از موجک های Daubechies، در میان محبوب ترین موج های شکل برداری، در شکل ۸.۴۳ (b) استفاده شده است، و symlets، که امتداد موجک های Daubechies با افزایش تقارن هستند، در شکل ۸.۴۳ (c) استفاده شده است. موجک های Cohen-Daubechies-Feauveau که در شکل ۸.۴۳ (d) به کار گرفته شده اند، برای نشان دادن قابلیت های موجک های دوضلعی گنجانده شده است. همانطور که در نتایج قبلی این نوع، همه ضرایب جزئیات مقیاس بندی شدند تا ساختار زیرین بیشتر نمایان شود، با شدت ۱۲۸ مربوط به مقدار ضریب صفر. همانطور که در جدول ۸.۱۴ مشاهده می کنید، تعداد عملیات مربوط به محاسبه تبدیل ها در شکل ۸.۴۳ از ۴ تا ۲۸ ضرب و اضافه در هر ضریب (برای هر تجزیه افزایش می یابد).

جدول ۸.۱۴ شکل شیرهای فیلتر و ضرایب صفر زمانی که در شکل ۸.۴۳ تبدیل می کند

ضرایب فیلتر  
(مقیاس بندی + موجک)

موجک	Filter Taps (Scaling + Wavelet)	ضرایب صفر
Haar	2 + 2	33.3%
Daubechies	8 + 8	40.9%
Symlet	8 + 8	41.2%
Biorthogonal	17 + 11	42.1%



شکل ۸.۴۳ موج سه مقیاس اجازه دهید شکل ۸.۹ (a) نسبت به (الف) موجک های Haar ، (ب) موجک های Daubechies ، (ج) سملت ها و (د) کوهن- موجک های دو وجهی Daubechies-Feau-auau . سطح هنگام حرکت از شکل ۸.۴۳ (a) به (d). هر چهار تبدیل با استفاده از فرمول تبدیل موجک سریع (به عنوان مثال ، بانک فیلتر) محاسبه شد. توجه داشته باشید که با افزایش پیچیدگی محاسباتی (یعنی تعداد ضربه های فیلتر) ، تابع بسته بندی اطلاعات نیز افزایش می یابد. وقتی از موجک های Haar استفاده می شود و ضرایب جزئیات زیر ۱.۵ به صفر می رسد ، ۳۳.۸٪ از کل تبدیل صفر می شود. با موجک های مختلط تر دو ضلعی ، تعداد ضرایب صفر شده به ۴۲.۱ rises افزایش می یابد و فشرده سازی بالقوه را تقریباً ۱۰ افزایش می دهد.

### انتخاب سطح تجزیه

یکی دیگر از عوامل موثر بر پیچیدگی محاسباتی کدگذاری موجک و خطای بازسازی ، تعداد سطوح تجزیه تبدیل است. از آنجا که تبدیل موجک مقیاس P شامل تکرارهای فیلتر P فیلتر است ، تعداد عملیات در محاسبه تبدیلهای رو به جلو و معکوس با تعداد سطوح تجزیه افزایش می یابد. علاوه بر این ، کمی کردن مقیاس پایین تر به طور فزاینده ای ضرایبی که با سطوح تجزیه بیشتر منجر می شوند بر مناطق بزرگتر شکل بازسازی شده تأثیر می گذارد. در بسیاری از برنامه ها ، مانند جستجوی پایگاه داده های شکل یا ارسال تصاویر برای بازسازی پیشرونده ، وضوح تصاویر ذخیره شده یا منتقل شده و مقیاس کمترین تقریب های مفید ، به طور معمول تعداد سطوح تبدیل را تعیین می کند.

مثال ۸.۲۶: سطوح تجزیه در کدگذاری موجک.

جدول ۸.۱۵ تأثیر انتخاب سطح تجزیه را بر روی کدگذاری شکل ۸.۹ (الف) با استفاده از موجک های دو متعامد و یک آستانه جهانی ثابت ۲۵ نشان می دهد. همانطور که در مثال کدگذاری موجک قبلی ، فقط ضرایب جزئیات کوتاه شده اند. این جدول هم درصد ضرایب صفر شده و هم خطاهای بازسازی rms را از معادله (۸-۱۰) نشان می دهد. توجه داشته باشید که تجزیه های اولیه مسئول بیشتر فشردن داده ها هستند. تبدیلات کمی در تعداد ضرایب ضخیم بالای سه سطح تجزیه وجود دارد.

### طراحی کوآنتایزر

مهمترین عامل موثر بر فشردن سازی و خطای بازسازی موجک ، کوآنتیزاسیون ضریب است. اگرچه پرکاربردترین کوآنتایزرها یکنواخت هستند ، اما می توان اثر کوآنتیزاسیون را به میزان قابل توجهی بهبود بخشید (۱) معرفی یک فاصله کوآنتیزاسیون بزرگتر در حدود صفر ، به نام منطقه مرده ، یا (۲) تطبیق اندازه فاصله کوآنتیزاسیون از مقیاس به مقیاس. در هر صورت ، فواصل کوآنتیزاسیون انتخاب شده باید با جریان بیت رمزگذاری شده شکل به رمزگشایی منتقل شود. فواصل ممکن است به صورت اکتشافی تعیین شوند یا به طور خودکار بر اساس شکل فشردن شده محاسبه شوند. به عنوان مثال ، یک آستانه ضریب جهانی می تواند به عنوان میانگین مقادیر مطلق ضرایب جزئیات سطح اول یا به عنوان تابعی از تعداد صفرهایی که کوتاه شده اند و مقدار انرژی ذخیره شده در شکل بازسازی شده محاسبه شود.

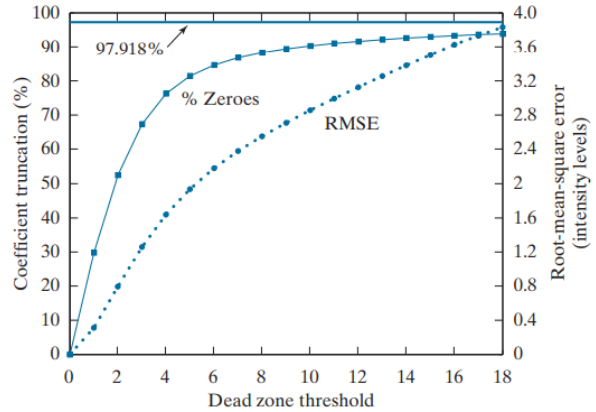
یکی از معیارهای اندازه گیری انرژی یک سیگنال دیجیتالی مجموع نمونه های مربع است.

مثال ۸.۲۷: انتخاب فاصله منطقه مرده در کدگذاری موجک.

شکل ۸.۴۴ تأثیر اندازه فاصله بین مناطق مرده را بر درصد ضرایب مختصر جزئی برای کدگذاری سه ضلعی بر اساس موجک سه مقیاس شکل ۸.۹ (الف) نشان می دهد. با افزایش اندازه منطقه مرده ، تعداد ضرایب کوتاه شده نیز افزایش می یابد. بالای زانوی منحنی (یعنی بیش از ۵) ،

جدول ۸.۱۵ تأثیر سطح تجزیه بر موجک کد کننده شکل ۵۱۲\*۵۱۲ از شکل ۸.۹ (الف) a)

سطح تجزیه (مقیاس یا تکرار بانک فیلتر)	ضریب تقریب تصویر	ضرایب کوتاه شده (%)	خطای بازسازی (rms)
1	256 × 256	74.7%	3.27
2	128 × 128	91.7%	4.23
3	64 × 64	95.1%	4.54
4	32 × 32	95.6%	4.61
5	16 × 16	95.5%	4.63



شکل ۸.۴۴ تاثیر انتخاب فاصله منطقه مرده بر روی کدگذاری موجک

بهره کمی وجود دارد این به این دلیل است که هیستوگرام ضرایب جزئیات در حدود صفر است. خطاهای بازسازی rms مربوط به آستانه های منطقه مرده در شکل ۸.۴۴ از ۰ تا ۱.۹۴ سطح شدت در آستانه ۵ و تا ۳.۸۳ سطح شدت برای آستانه ۱۸ افزایش می یابد، جایی که تعداد صفرها به ۹۳.۸۵ reaches می رسد. اگر همه ضرایب جزئی حذف شوند، این درصد به حدود ۹۷.۹۲ افزایش (حدود ۴) افزایش می یابد، اما خطای بازسازی تا ۱۲.۳ سطح شدت افزایش می یابد.

### JPEG-۲۰۰۰

JPEG-۲۰۰۰ استاندارد JPEG را گسترش می دهد تا انعطاف پذیری بیشتری را در فشرده سازی تصاویر ثابت با صدای ثابت و دسترسی به داده های فشرده افزایش دهد. به عنوان مثال، بخش هایی از یک شکل فشرده JPEG-۲۰۰۰ را می توان برای ارسال مجدد، ذخیره، نمایش و/یا ویرایش استخراج کرد. این استاندارد بر اساس تکنیک های کدگذاری موجک است که توضیح داده شد. کوانتیزه ضریب با مقیاس ها و زیر باند های جداگانه سازگار است و ضرایب کوانتیزه به صورت حسابی بر اساس سطح بیتی کدگذاری می شوند (به بخش های ۸.۴ و ۸.۸ مراجعه کنید). با استفاده از نماد استاندارد، یک شکل به صورت زیر کدگذاری می شود (ISO/IEC [۲۰۰۰]).

اولین مرحله از فرآیند کدگذاری این است که نمونه های شکل بدون علامت  $Ss_{iz}$ -bit را با تفریق  $2^{Ss_{iz}-1}$  در سطح DC تبدیل دهید. اگر شکل دارای بیش از یک جزء باشد، مانند صفحات قرمز، سبز و آبی یک شکل رنگی، هر جزء به صورت جداگانه جابجا می شود. اگر دقیقاً سه جزء وجود داشته باشد، ممکن است به صورت اختیاری با استفاده از یک ترکیب خطی برگشت پذیر یا غیرقابل برگشت از اجزاء از هم مرتبط شوند. به عنوان مثال، تبدیل اجزای برگشت ناپذیر استاندارد است

$$\begin{aligned}
 Y_0(x, y) &= 0.299I_0(x, y) + 0.587I_1(x, y) + 0.114I_2(x, y) \\
 Y_1(x, y) &= -0.16875I_0(x, y) - 0.33126I_1(x, y) + 0.5I_2(x, y) \\
 Y_2(x, y) &= 0.5I_0(x, y) - 0.41869I_1(x, y) - 0.08131I_2(x, y)
 \end{aligned} \tag{8-60}$$

جایی که ۱۰، ۱۱ و ۱۲ اجزای ورودی با تبدیل سطح هستند و ۷۰، ۷۱ و ۷۲ اجزای مربوط به هم مرتبط نیستند. اگر اجزای ورودی صفحات قرمز، سبز و آبی یک شکل رنگی هستند، معادله (۸-۶۰) تقریباً تبدیل ویدئوی رنگی  $R'G'B'$  به  $YCbCr$  (Poynton [۱۹۹۶]) هدف از تبدیل، بهبود کارایی فشرده سازی است. اجزای تبدیل یافته ۷۱ و ۷۲ تفاوت دارند. تصاویری که هیستوگرام آنها بسیار صفر است.

پس از تبدیل سطح شکل و ارتباط اختیاری، اجزای آن را می توان به کاشی تقسیم کرد. کاشی ها آرایه های مستطیلی از پیکسل ها هستند که به طور مستقل پردازش می شوند. از آنجا که یک شکل می تواند بیش از یک جزء داشته باشد (به عنوان مثال، می تواند از سه جزء رنگی تشکیل شده باشد)، فرایند کاشی کاری باعث ایجاد اجزای کاشی می شود. هر جزء کاشی را می توان به طور مستقل بازسازی کرد و مکانیزمی ساده برای دسترسی و/یا دستکاری در محدوده محدودی از یک شکل کدگذاری شده ارائه داد. به عنوان مثال، شکل با نسبت شکل ۱۶:۹ می تواند به کاشی تقسیم شود، بنابراین یکی از کاشی های آن یک شکل فرعی با نسبت شکل ۴:۳ است. سپس می توان آن کاشی را بدون دسترسی به کاشی های دیگر در شکل فشرده بازسازی کرد. اگر شکل به کاشی تقسیم نشده باشد، یک کاشی واحد است.

اندازه  $S$  در استاندارد برای نشان دادن وضوح شدت استفاده می شود. تبدیل اجزای برگشت ناپذیر، تبدیل جزء مورد استفاده برای تراکم تلفاتی است. تبدیل جزء به خودی خود برگشت ناپذیر نیست. متفاوتی برای فشرده سازی برگشت پذیر استفاده می شود

سپس تبدیل موجک گسسته ۱ بعدی ردیفها و ستونهای هر جزء کاشی محاسبه می شود. برای فشرده سازی بدون خطا، تبدیل بر اساس یک مقیاس بندی ضریب و  $3/5$  ضریب و موجک انجام می شود (Le Gall and [۱۹۸۸]). یک روش گرد کردن برای ضرایب تبدیل بدون عدد صحیح تعریف شده است. در برنامه های زیان آور، یک بردار موجک ضخیم  $7/9$  استفاده می شود (آنتونینی، بارلاود، ماتیو و داوچی [۱۹۹۲]). در هر صورت، تبدیل با استفاده از تبدیل موجک سریع بخش  $7.10$  یا از طریق رویکرد مکمل بر اساس بلند کردن (Mallat [۱۹۹۹]) محاسبه می شود. به عنوان مثال، در کاربردهای زیان آور، ضرایب مورد استفاده برای ساخت بانک فیلتر تجزیه و تحلیل  $9/7$  FWT در جدول  $7.1$  آورده شده است. اجرای تکمیلی مبتنی بر بلند کردن شامل شش عملیات متوالی "بلند کردن" و "مقیاس بندی" است:

$$\begin{aligned} Y(2n+1) &= X(2n+1) + \alpha[X(2n) + X(2n+2)] & i_0 - 3 \leq 2n+1 < i_1 + 3 \\ Y(2n) &= X(2n) + \beta[Y(2n-1) + Y(2n+1)] & i_0 - 2 \leq 2n < i_1 + 2 \\ Y(2n+1) &= Y(2n+1) + \gamma[Y(2n) + Y(2n+2)] & i_0 - 1 \leq 2n+1 < i_1 + 1 \\ Y(2n) &= Y(2n) + \delta[Y(2n-1) + Y(2n+1)] & i_0 \leq 2n < i_1 \\ Y(2n+1) &= -KY(2n+1) & i_0 \leq 2n+1 < i_1 \\ Y(2n) &= Y(2n)/K & i_0 \leq 2n < i_1 \end{aligned}$$

(8-61)

پیاده سازی های مبتنی بر لیفتینگ روش دیگری برای محاسبه تبدیل موجک است. ضرایب مورد استفاده در این رویکرد مستقیماً به ضرایب بانک FWT مربوط است.

در اینجا،  $X$  جزء کاشی است که در حال تبدیل است،  $Y$  تبدیل حاصله است و  $i_0$  و  $i_1$  موقعیت جزء کاشی را در یک جزء مشخص می کنند. یعنی، آنها شاخص های اولین نمونه از ردیف یا ستون جزء کاشی در حال تبدیل و نمونه بلافاصله پس از آخرین نمونه هستند. متغیر  $n$  مقادیر را بر اساس  $i_0$ ،  $i_1$  فرض می کند و تعیین می کند که کدام یک از شش عملیات در حال انجام است. اگر  $n < i_0$  or  $n > i_1$  باشد،  $X(n)$  با گسترش متقارن بدست می آید. به عنوان مثال،  $X(i_0 - 1) = X(i_0 + 1)$ ،  $X(i_0 - 2) = X(i_0 + 2)$ ،  $X(i_1) = X(i_1 - 2)$  و  $X(i_1 + 1) = X(i_1 - 3)$  در پایان عملیات بلند کردن و مقیاس بندی، مقادیر یکنواخت  $Y$  معادل خروجی فیلتر شده FWT پایین گذر است. عجیب-مقادیر نمایه شده از  $Y$  با نتیجه فیلتر شده FWT بالا گذر مطابقت دارد. پارامترهای  $a, b, g$  و  $d = -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45$ ، به ترتیب، و ضریب مقیاس بندی  $K = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45$  است.

این ضرایب مبتنی بر بلند کردن در استاندارد مشخص شده اند. از فصل ۷ به یاد بیاورید که DWT شکل را به مجموعه ای از اجزای محدود باند موسوم به زیر باند تجزیه می کند.

دگرگونی که به تازگی توضیح داده شد، چهار زیر باند ایجاد می کند. تقریبی با وضوح پایین از اجزای کاشی و ویژگی های فرکانس افقی، عمودی و مورب آن. تکرار بار تبدیل NL، با تکرارهای بعدی محدود به ضرایب تقریبی تجزیه قبلی، تبدیل موجک در مقیاس NL ایجاد می کند. مقیاس های مجاور با قدرت ۲ به هم مرتبط هستند و پایین ترین مقیاس شامل تنها تقریب صریحاً مشخص شده از جزء اصلی کاشی است. همانطور که از شکل ۸.۴۵ قابل تصور است، جایی که نماد استاندارد JPEG-۲۰۰۰ برای مورد  $NL = 2$  خلاصه شده است، یک تبدیل کلی در مقیاس NL شامل زیربندهای  $NL-13$  است که ضرایب ضریب آنها برای  $b = NL HL$  نشان داده شده است، ...،  $HL_1$ ،  $LH_1$ ،  $HH_1$ . استاندارد تعداد مقیاس های مورد محاسبه را مشخص نمی کند.

هنگامی که هر یک از اجزای کاشی پردازش می شوند، تعداد کل ضرایب تبدیل برابر با تعداد نمونه های موجود در شکل اصلی است، اما اطلاعات بصری مهم در چند ضخامت متمرکز می شود. برای کاهش تعداد بیت های مورد نیاز برای نمایش تبدیل، ضریب  $a_b(u, v)$  زیر باند  $b$  به مقدار  $q_b(u, v)$  با استفاده از

$$q_b(u, v) = \text{sign}[a_b(u, v)] \cdot \text{floor} \left[ \frac{|a_b(u, v)|}{\Delta_b} \right] \quad (8-62)$$

جایی که اندازه مرحله اندازه گیری  $\Delta_b$  در آن است

$$\Delta_b = 2^{R_b - \epsilon_b} \left( 1 + \frac{\mu_b}{2^{11}} \right) \quad (8-63)$$

تعریف استاندارد محدوده دینامیکی اسمی را با تعریف نزدیک در فصل ۲ اشتباه نگیرد.

Rb محدوده پویای اسمی زیر باند b است ، در حالی که eb و mb تعداد بیت هایی است که به ضریب و مانیتیسای ضرایب زیر باند اختصاص داده شده است. محدوده پویای اسمی زیر باند b مجموع تعداد بیت هایی است که برای نشان دادن شکل اصلی و بیت های تجزیه و تحلیل برای زیر باند b استفاده می شود. بیت های تجزیه و تحلیل زیر باند از الگوی ساده نشان داده شده در شکل ۸.۴۵ پیروی می کنند. به عنوان مثال ، دو بیت افزایش تجزیه و تحلیل برای زیر باند HH = ۱b وجود دارد.

برای فشرده سازی بدون خطا ،  $Rb = eb, mb - 0$  و  $\Delta b = 1$  برای فشار غیر قابل برگشت ، هیچ اندازه مرحله کوانتاسیون خاصی در استاندارد مشخص نشده است. بجای، تعداد بیت های نمایی و مانیتیسای باید بر اساس زیر باند در اختیار رمزگشایی قرار گیرد که به آن کوانتیزاسیون تفسیری گفته می شود ، یا فقط برای زیر باند NLLL ، که کوانتیزاسیون مشتق شده نامیده می شود. در حالت دوم ، باند فرعی باقی مانده با استفاده از پارامترهای زیر باند NLLL برون یابی شده ، کوانتیزه می شوند. اگر  $e_0$  و  $m_0$  تعداد بیت های اختصاص یافته به زیر باند NLLL باشند ، پارامترهای برآمده برای زیر باند b عبارتند از:

$$\begin{aligned} \mu_b &= \mu_0 \\ \varepsilon_b &= \varepsilon_0 + n_b - N_L \end{aligned} \quad (8-64)$$

جایی که nb تعداد سطوح تجزیه زیر باند را از جزء اصلی کاشی شکل به زیر باند نشان می دهد در مراحل نهایی فرآیند کدگذاری ، ضرایب هر زیربند تبدیل شده به جزء کاشی در بلوک های مستطیلی به نام بلوک های کد قرار می گیرند که به صورت جداگانه کدگذاری می شوند ، یک صفحه بیت در یک زمان. با شروع از مهمترین صفحه بیت با یک عنصر غیر صفر ، هر صفحه بیت در سه مرحله پردازش می شود. هر بیت (در صفحه بیت) تنها در یکی از سه گذر رمزگذاری شده است که به آنها انتشار معنادار ، بازآرایی قدر و پاکسازی گفته می شود.

سپس خروجی ها به صورت حسابی کدگذاری شده و با گذرهای مشابه از سایر بلوک های کد گروه بندی می شوند تا لایه ها را تشکیل دهند. یک لایه عبارت است از تعداد دلخواه گروه بندی رمزهای عبور از هر بلوک کد. لایه های به وجود آمده در نهایت به بسته ها تقسیم می شوند و یک روش اضافی برای استخراج یک منطقه فضایی مورد علاقه از کل جریان کد ارائه می دهند. بسته ها واحد اساسی جریان کدگذاری شده هستند. رمزگشایی JPEG-۲۰۰۰ به سادگی عملیات قبلی را وارونه می کند. پس از بازسازی زیربندهای اجزای کاشی از بسته های JPEG-۲۰۰۰ که به صورت حسابی کدگذاری شده اند ، تعدادی از زیربندهای انتخاب شده توسط کاربر رمزگشایی می شود.

اگرچه رمزگذار ممکن است سطوح بیتی Mb را برای یک زیر باند خاص رمزگذاری کرده باشد ، اما کاربر ، به دلیل ماهیت جاسازی شده جریان کد ، می تواند رمزگشایی فقط سطوح بیتی Nb را انتخاب کند. این به معنی کمی کردن ضرایب بلوک کد با استفاده از یک مرحله است



$a_{2LL}(u, v)$ 0	$a_{2HL}(u, v)$ 1	$a_{1HL}(u, v)$ 1
$a_{2LH}(u, v)$ 1	$a_{2HH}(u, v)$ 2	
$a_{1LH}(u, v)$ 1		$a_{1HH}(u, v)$ 2

شکل ۸.۴۵ JPEG ۲۰۰۰ دو-تبدیل موجک مقیاس کاشی-اجزای سازنده نماد و افزایش تجزیه و تحلیل. حجم ۲ مگابایت  $2^{M_b-N_b} \cdot \Delta_b$  هر بیت غیر رمزگذاری شده بر روی صفر تنظیم می شود و ضرایب حاصله ، که با  $q_b(u, v)$  مشخص می شوند ، با استفاده از

$$R_{q_b}(u, v) = \begin{cases} \left( \bar{q}_b(u, v) + r \cdot 2^{M_b-N_b(u, v)} \right) \cdot \Delta_b & \bar{q}_b(u, v) > 0 \\ \left( \bar{q}_b(u, v) - r \cdot 2^{M_b-N_b(u, v)} \right) \cdot \Delta_b & \bar{q}_b(u, v) < 0 \\ 0 & \bar{q}_b(u, v) = 0 \end{cases} \quad (8-65)$$

تعیین مقدار همانطور که در قسمت قبل تعریف شده است برگشت ناپذیر است. اصطلاح "کمیت معکوس" به این معنا نیست که هیچ گونه اطلاعاتی از دست نمی رود. این فرایند به جز در مورد فشرده سازی برگشت پذیر JPEG-۲۰۰۰ ضرر دارد ، جایی که  $Rb = eb$  ،  $mb = 0$  و  $\Delta b = 1$  است.

جایی که  $R_{q_b}(u, v)$  نشان دهنده ضریب تبدیل معکوس شده است و  $N_b(u, v)$  تعداد صفحات بیت رمزگشایی شده برای  $q_b(u, v)$  است. پارامتر ۲ بازسازی توسط رمزگشایی انتخاب می شود تا بهترین کیفیت بصری یا عینی بازسازی را تولید کند. به طور کلی ،  $0 \leq r < 1$  ، با مقدار مشترک  $r = 1/2$  . ضرایب معکوس کمی شده توسط ستون و ردیف با استفاده از یک بانک فیلتر  $FWT^{-1}$  که ضرایب آن از جدول ۷.۱ بدست آمده است ، معکوس می شوند. یا از طریق عملیات زیر بر اساس بلند کردن:

$$\begin{array}{ll} X(2n) = K \cdot Y(2n) & i_0 - 3 \leq 2n < i_1 + 3 \\ X(2n+1) = (-1/K)Y(2n+1) & i_0 - 2 \leq 2n - 1 < i_1 + 2 \\ X(2n) = X(2n) - \delta[X(2n-1) + X(2n+1)] & i_0 - 3 \leq 2n < i_1 + 3 \\ X(2n+1) = X(2n+1) - \gamma[X(2n) + X(2n+2)] & i_0 - 2 \leq 2n + 1 < i_1 + 2 \\ X(2n) = X(2n) - \beta[X(2n-1) + X(2n+1)] & i_0 - 1 \leq 2n < i_1 + 1 \\ X(2n+1) = X(2n+1) - \alpha[X(2n) + X(2n+2)] & i_0 \leq 2n + 1 < i_1 \end{array} \quad (8-66)$$

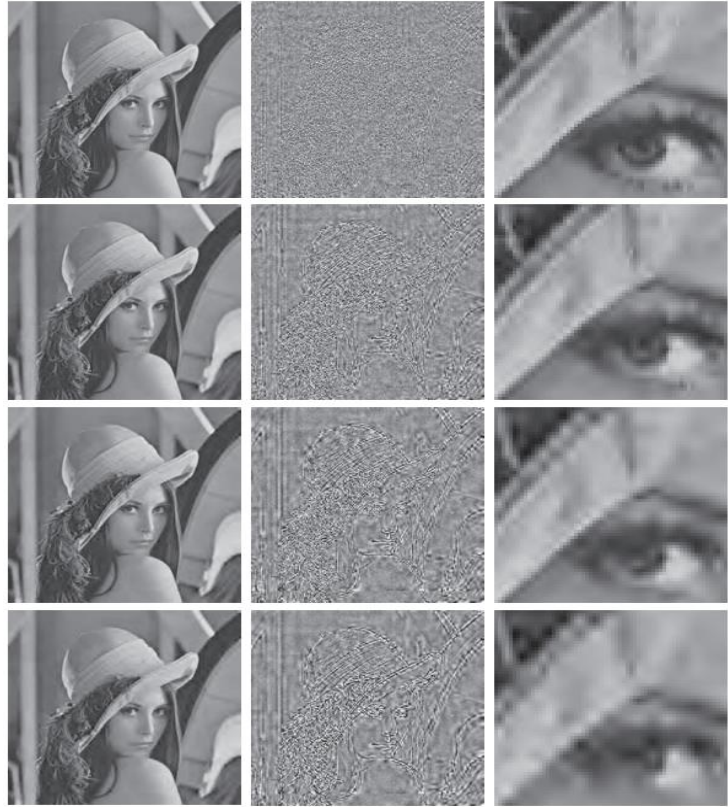
که در آن پارامترهای  $a, b, g, d$  و  $K$  مطابق معادله (۸-۶۱) تعریف شده است. ردیف ضریب معکوس یا عنصر ستون  $Y(n)$  در صورت لزوم به طور متقارن گسترش می یابد. مراحل رمزگشایی نهایی عبارتند از مونتاژ کاشی های اجزاء، تبدیل اجزای معکوس (در صورت نیاز) و تبدیل سطح DC. برای کدگذاری برگشت ناپذیر، تبدیل جزء معکوس است

$$\begin{aligned} I_0(x, y) &= Y_0(x, y) + 1.402Y_2(x, y) \\ I_1(x, y) &= Y_0(x, y) - 0.34413Y_1(x, y) - 0.71414Y_2(x, y) \\ I_2(x, y) &= Y_0(x, y) + 1.772Y_1(x, y) \end{aligned} \quad (8-67)$$

و پیکسل های تبدیل شده توسط  $2^{Ssz-1}$  تبدیل مکان می دهند.

مثال ۸.۲۸: مقایسه کدگذاری مبتنی بر موجک JPEG-۲۰۰۰ و فشرده سازی مبتنی بر DCT JPEG.

شکل ۸.۴۶ چهار تقریب JPEG-۲۰۰۰ از شکل تک رنگ را در شکل ۸.۹ (a) نشان می دهد. سطرهای بعدی شکل افزایش سطح فشرده سازی را نشان می دهد، از جمله  $C = ۲۵$ ،  $۵۲$ ،  $۷۵$  و  $۱۰۵$ . تصاویر در ستون ۱ رمزگذاری JPEG-۲۰۰۰ را فشرده می کنند. تفاوت بین این تصاویر و شکل اصلی [شکل ۸.۹ (الف) را ببینید] در ستون دوم نشان داده شده است، و ستون سوم شامل یک قسمت بزرگنمایی شده از بازسازی ها در ستون ۱ است. زیرا نسبت های فشرده سازی برای دو ردیف اول عبارتند از: تقریباً با نسبت های فشرده سازی در مثال ۸.۱۸، این نتایج را می توان (از نظر کمی و کیفی) با نتایج مبتنی بر تبدیل JPEG در شکل ۸.۲۹ (الف) تا (و) مقایسه کرد.



شکل ۸.۴۶ چهار تقریب JPEG-۲۰۰۰ از شکل ۸.۹ (الف). هر سطر شامل یک نتیجه پس از فشرده سازی و بازسازی ، تفاوت مقیاس شده بین نتیجه و شکل اصلی و یک قسمت بزرگنمایی شده از شکل بازسازی شده است. (نتایج ردیف ۱ و ۲ را با نتایج JPEG در شکل ۸.۲۹ مقایسه کنید).

مقایسه بصری تصاویر خطا در ردیف ۱ و ۲ شکل ۸.۴۶ با تصاویر مربوطه در شکل ۸.۲۹ (b) و (e) کاهش قابل توجهی از خطا را در نتایج JPEG-۲۰۰۰ نشان می دهد-۳.۸۶ و ۵.۷۷ سطح شدت ، در مقابل ۵.۴ و ۱۰.۷ سطح شدت برای نتایج JPEG خطاهای محاسبه شده بر اساس نتایج موجک در هر دو سطح فشرده سازی مطلوب است. علاوه بر کاهش خطای بازسازی ، برنامه نویسی موجک به طور چشمگیری (در مفهوم ذهنی) کیفیت شکل را افزایش می دهد. توجه داشته باشید که مساله مسدودکننده بر نتایج JPEG تسلط دارد [شکلها را ببینید. ۸.۲۹ (c) و (f)] در شکل ۸.۴۶ وجود ندارد. در نهایت ، توجه داریم که فشرده سازی در ردیف ۳ و ۴ شکل ۸.۴۶ با JPEG عملی نیست. JPEG-۲۰۰۰ تصاویری قابل استفاده ارائه می دهد که بیش از ۱:۱۰۰ فشرده می شوند ، و قابل توجه ترین تخریب افزایش تاری شکل است.

## ۸.۱۲ برچسب گذاری شکل دیجیتالی

روشها و استانداردهای بخشهای ۸.۲ تا ۸.۱۱ باعث توزیع تصاویر (در عکسها یا فیلمها) در رسانه های دیجیتالی و از طریق اینترنت می شود. متأسفانه ، تصاویری که به این ترتیب توزیع می شوند می توانند مکرراً و بدون خطا کپی شوند و حقوق صاحبان آنها را در معرض خطر قرار دهند. حتی وقتی برای توزیع رمزگذاری می شوند ، تصاویر

پس از رمزگشایی محافظت نمی شوند. یکی از راههای جلوگیری از تکرار غیرقانونی این است که یک یا چند مورد از اطلاعات را که در مجموع علامت نامیده می شود ، در تصاویر بالقوه آسیب پذیر قرار دهید ، به گونه ای که علامت های آب از خود تصاویر جدا نشدنی باشند. آنها به عنوان بخشهای جدایی ناپذیر تصاویر برچسب گذاری شده ، از حقوق صاحبان خود به طرق مختلف محافظت می کنند ، از جمله:

۱. شناسایی حق چاپ. علامت گذار می تواند اطلاعاتی را ارائه دهد که به عنوان اثبات مالکیت در صورت نقض حقوق مالک عمل می کند.

۲. شناسایی کاربر یا چاپ انگشت. می توان یک کاربران حقوقی را در علامت های آبی کدگذاری کرد و از آنها برای شناسایی منابع نسخه های غیرقانونی استفاده کرد.

۳. تعیین اصالت. وجود علامت می تواند تضمین کننده عدم تبدیل شکل باشد ، با این فرض که علامت به گونه ای طراحی شده است که با هرگونه تبدیل در شکل از بین برود.

۴. نظارت خودکار. علامت های آب را می توان با سیستم هایی که زمان و مکان استفاده از تصاویر را ردیابی می کنند (به عنوان مثال برنامه هایی که در وب برای جستجوی تصاویر قرار گرفته در صفحات وب جستجو می کنند) نظارت کرد. نظارت برای جمع آوری حق امتیاز و/یا موقعیت کاربران غیرقانونی مفید است.

۵. حفاظت از کپی. علامت گذار ها می توانند قوانین استفاده و کپی شکل (به عنوان مثال ، در دستگاه های پخش DVD) را مشخص کنند.

در این بخش ، ما یک مرور کوتاه از برچسب گذاری شکل دیجیتال ارائه می دهیم ، که عبارت است از قرار دادن داده ها در یک شکل به گونه ای که می توان از آنها برای اظهار نظر در مورد شکل استفاده کرد. روشهای توصیف شده با تکنیکهای فشرده سازی ارائه شده در بخشهای قبلی (اگرچه شامل کدگذاری اطلاعات هستند) شباهت چندانی ندارند. در حقیقت ، علامت نگاری و فشرده سازی از برخی جهات متضاد هستند. در حالی که هدف در فشرده سازی کاهش داده های مورد استفاده برای نمایش تصاویر است ، هدف در برچسب گذاری برچسب گذاری اطلاعات و داده ها (یعنی علامت های علامت گذار ) به آنها است. همانطور که در قسمت های دیگر نشان داده می شود ، علامت های خود می توانند یا قابل مشاهده یا نامرئی باشند.

علامت گذار قابل مشاهده یک زیر شکل یا شکل مات یا نیمه شفاف است که بر روی شکل دیگری (به عنوان مثال ، شکل در حال برچسب گذاری ) قرار می گیرد تا برای بیننده آشکار باشد. شبکه های تلویزیونی اغلب علامت های قابل مشاهده (که بر اساس نشان های آنها طراحی شده است) را در گوشه سمت راست بالا یا پایین صفحه تلویزیون قرار می دهند. همانطور که مثال زیر نشان می دهد ، برچسب گذاری قابل مشاهده معمولاً در حوزه فضایی انجام می شود.

مثال ۸.۲۹: یک علامت گذار ساده قابل مشاهده.

شکل در شکل ۸.۴۷ (ب) چهار ضلع پایین سمت راست شکل در شکل ۸.۹ (الف) است که نسخه مقیاس بندی شده علامت در شکل ۸.۴۷ (الف) روی آن پوشانده شده است. اگر  $f_w$  نشان دهنده شکل برچسب گذاری شده باشد، می توانیم آن را به صورت ترکیبی خطی از شکل بدون علامت  $f$  و علامت  $w$  با استفاده از

$$f_w = (1 - \alpha)f + \alpha w \quad (8-68)$$

جایی که ثابت  $\alpha$  دید نسبی علامت و شکل زیرین را کنترل می کند. اگر  $\alpha = 1$  باشد، علامت گذار مات است و شکل زیرین کاملاً مبهم است. با نزدیک شدن به  $0$ ، بیشتر شکل زیرین و کمتر علامت دیده می شود. به طور کلی،  $0 < \alpha \leq 1$  در شکل ۸.۴۷ (ب)،  $\alpha = 0.3$ ، شکل ۸.۴۷ (ج) تفاوت محاسبه شده (از نظر شدت) بین شکل برچسب گذاری شده در (ب) و شکل بدون علامت در شکل ۸.۹ (الف) است. شدت ۱۲۸ نشان دهنده تفاوت  $0$  است. توجه داشته باشید که شکل زیر از طریق علامت "نیمه شفاف" به وضوح قابل مشاهده است. این موضوع هم در شکل ۸.۴۷ (ب) و هم در شکل تفاوت در شکل ۸.۴۷ (ج) مشهود است. برخلاف علامت گذار قابل مشاهده در مثال قبلی، علامت های نامرئی با چشم غیر مسلح قابل مشاهده نیستند. آنها نامحسوس هستند اما می توان آنها را با یک الگوریتم رمزگشایی مناسب بازیابی کرد. نامرئی شدن با قرار دادن آنها به عنوان اطلاعات بصری اضافی (اطلاعاتی که سیستم بینایی انسان نادیده می گیرد یا نمی تواند

## Digital Image Processing



شکل ۸.۴۷ یک علامت گذار ساده قابل مشاهده: (الف) علامت؛ (ب) شکل مشخص شده با  $\alpha = 0.3$  و (ج) تفاوت بین شکل برچسب گذاری شده با شکل اصلی (بدون برچسب گذاری نشده). درک کنید (بخش ۸.۱ را ببینید). شکل ۸.۴۸ (الف) یک مثال ساده ارائه می دهد. از آنجا که کمترین مقدار قابل توجهی از یک شکل ۸ بیتی عملاً هیچ تاثیری در درک ما از شکل ندارد، علامت نمایی از شکل ۸.۴۷ (a) در دو بیت کم اهمیت آن وارد شده یا "پنهان" شده است. با استفاده از نماد معرفی شده در بالا، اجازه می دهیم

$$f_w = 4 \left( \frac{f}{4} \right) + \frac{w}{64} \quad (8-69)$$

و از محاسبات صحیح بدون علامت برای انجام محاسبات استفاده کنید. تقسیم و ضربدر ۴ بر روی دو بیت کمترین میزان  $f$  را به ۰ می دهد ، تقسیم  $w$  بر ۶۴ دو بیت مهم آن را به دو موقعیت بیتی کم اهمیت منتقل می کند و با افزودن دو نتیجه ، شکل برچسب گذاری LSB ایجاد می شود. توجه داشته باشید که علامت جاسازی شده در شکل ۸.۴۸ (الف) قابل مشاهده نیست. با به صفر رساندن مهمترین ۶ بیت این شکل و مقیاس گذاری مقادیر باقی مانده در محدوده شدت کامل ، می توان علامت آب را مانند شکل ۸.۴۸ (ب) استخراج کرد.

یکی از ویژگیهای مهم علامتهای نامرئی مقاومت آنها در برابر تلاشهای تصادفی و عمدی برای حذف آنها است. علامت های نامرئی شکننده با هرگونه تبدیل در تصاویر که در آنها جاسازی شده است از بین می روند. در برخی از برنامه ها ، مانند احراز یکه شکل ، این یک ویژگی مطلوب است. همانطور که در شکل ۸.۴۸ (ج) و (د) نشان می دهد ، شکل برچسب گذاری شده LSB در شکل ۸.۴۸ (الف) حاوی یک علامت نماد نامرئی شکننده است. اگر شکل (a) با استفاده از JPEG از دست رفته فشرده و بازسازی شود ، علامت از بین می رود. شکل ۸.۴۸ (ج) نتیجه پس از کام است



شکل ۸.۴۸ یک علامت نامرئی ساده: الف) شکل برچسب گذاری شده ؛ ب) علامت گذار استخراج شده ؛ ج) شکل مشخص شده از آب پس از فشرده سازی و فشرده سازی JPEG با کیفیت بالا ؛ و د) علامت آب استخراج شده از ج)

فشار دادن و فشرده سازی شکل ۸.۴۸ (a) ؛ خطای  $rms$  ۲٫۱ بیت است. اگر سعی کنیم علامت واترمارک را از این شکل با استفاده از روش مشابه (b) استخراج کنیم ، نتیجه غیرقابل درک است [شکل ۸.۴۸ (d) را ببینید]. اگرچه فشرده سازی و رفع فشار باعث از بین رفتن اطلاعات بصری مهم در شکل می شود ، اما علامت نازک شکننده از بین رفت. علامت های ویدئویی نامرئی قوی برای زنده ماندن از تبدیلات شکل طراحی شده اند ، خواه حملات به اصطلاح سهوی باشند یا عمدی. حملات غیرعادی رایج شامل فشرده سازی از دست دادن ، فیلترینگ

خطی و غیر خطی ، برش ، چرخش ، شکل برداری و موارد مشابه است. حملات عمدی از چاپ و اسکن مجدد گرفته تا افزودن علامت های اضافی و/یا نویز متغیر است. البته مقاومت در برابر حملات غیر ضروری است. که خود شکل را غیرقابل استفاده می گذارد.

شکل ۸.۴۹ اجزای اساسی یک سیستم برچسب گذاری عادی شکل را نشان می دهد. رمزگذار در شکل ۸.۴۹ (الف) علامت گذار  $w_i$  را در شکل  $fi$  قرار می دهد که باعث ایجاد شکل با علامت آب  $fw$  می شود. رمزگشای تکمیلی در (ب) وجود  $w_i$  را در ورودی برچسب گذاری شده  $fw$  یا ورودی بدون علامت  $fj$  استخراج و اعتبار می بخشد. اگر  $w_i$  قابل مشاهده است ، رمزگشایی لازم نیست در صورت نامرئی بودن ، رمزگشا ممکن است نیاز به کپی داشته باشد یا نداشته باشد.  $fi$  و  $w_i$  [در شکل ۸.۴۹ (ب) با رنگ آبی نشان داده شده است] تا کار خود را انجام دهد. در صورت استفاده از  $fi$  و/یا  $w_i$  ، سیستم برچسب گذاری به عنوان یک سیستم خصوصی یا کلید محدود شناخته می شود. در غیر این صورت ، یک سیستم عمومی یا کلید بدون محدودیت است. از آنجا که رمزگشا باید تصاویر برچسب گذاری شده و برچسب گذاری نشده را پردازش کند ،  $w$  در شکل ۸.۴۹ (ب) برای نشان دادن عدم وجود علامت استفاده می شود. در نهایت ، ما توجه داریم که برای تعیین وجود  $w_i$  در یک شکل ، رمزگشایی باید  $wj$  علامت استخراج شده را با  $w_i$  مرتبط کرده و نتیجه را با یک آستانه از پیش تعیین شده مقایسه کند. آستانه میزان شباهت قابل قبول برای "مسابقه" را تعیین می کند.

مثال ۸.۳۰: علامت گذار قوی نامرئی مبتنی بر DCT.

درج و استخراج علامت را می توان در حوزه مکانی ، مانند مثالهای قبلی ، یا در حوزه تبدیل انجام داد. شکل ۸.۵۰ (الف) و (ج) دو نسخه برچسب گذاری شده از شکل را در شکل ۸.۹ (الف) با استفاده از روش برچسب گذاری مبتنی بر DCT که در اینجا مشخص شده است نشان می دهد (کاکس و همکاران [۱۹۹۷]):

۱. DCT دو بعدی شکل را که قرار است برچسب گذاری شود ، محاسبه کنید.

۲.  $K$  بزرگترین ضرایب آن ،  $c_1$  ،  $c_2$  ، ... ،  $c_K$  را بر حسب قدر تعیین کنید.

۳. با ایجاد یک دنباله شبه تصادفی عنصر  $K$  ،  $v_1$  ،  $v_2$  ، ... ،  $v_K$  ، علامت نمایی ایجاد کنید که از توزیع گوسی با میانگین  $m = 0$  و واریانس  $\sigma^2 = 1$  گرفته شده است (توجه: یک عدد شبه تصادفی دنباله خواص اعداد تصادفی را تقریب می زند. این واقعاً تصادفی نیست زیرا به مقدار اولیه از پیش تعیین شده بستگی دارد).

۴- با استفاده از معادله زیر ، علامت مرحله ۳ را در  $K$  بزرگترین ضرایب DCT از مرحله ۲ قرار دهید.

$$c'_i = c_i \cdot (1 + \alpha v_i) \quad 1 \leq i \leq K \quad (8-70)$$

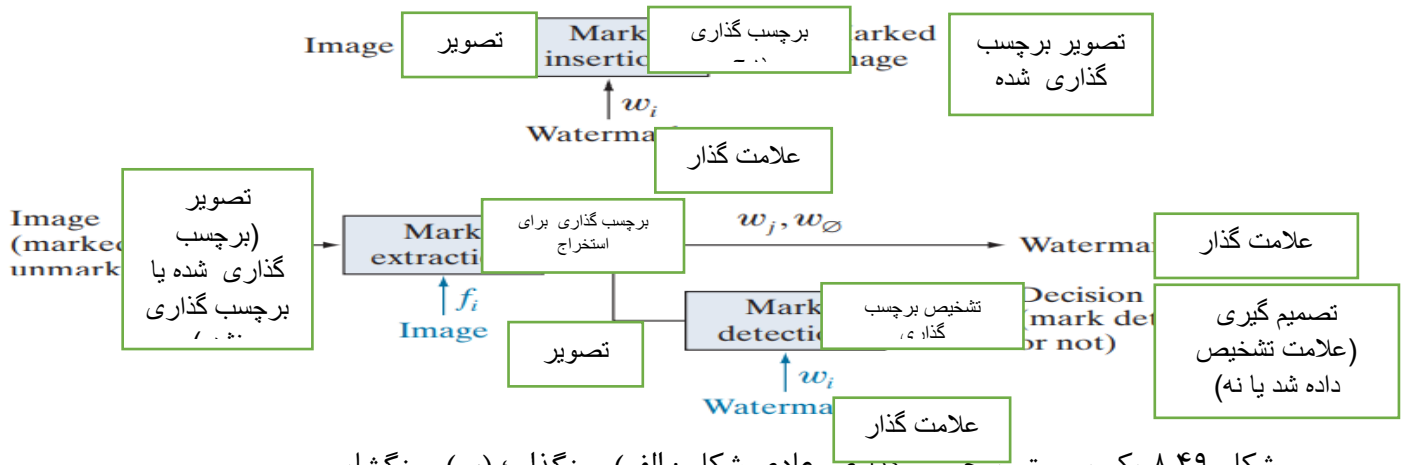
برای یک ثابت مشخص شده  $\alpha$  (که میزان تبدیل  $c_i$  را تبدیل می دهد). اصل را تعویض کنید.

$c_i$  با  $c'_i$  محاسبه شده از معادله (۸-۷۰). (برای تصاویر در شکل ۸.۵۰ ،  $\alpha = 0.1$  و  $K = 1000$ ).

۵. DCT معکوس نتیجه را از مرحله ۴ محاسبه کنید.



با استفاده از علامت گذار های ساخته شده از اعداد شبه تصادفی و پخش آنها بر روی اجزای فرکانس قابل ملاحظه شکل ، می توان  $a$  را کوچک کرد و دید علامت را کاهش داد. در عین حال ، امنیت علامت بالا نگه داشته می شود زیرا (۱) علامت های واترمار از اعداد شبه تصادفی و بدون ساختار مشخص تشکیل شده است ، (۲) علامت های علامت گذار در اجزای فرکانسی متعدد با تأثیر فضایی بر روی کل شکل دو بعدی ( بنابراین مکان آنها مشخص نیست) و (۳) حملات علیه آنها نیز باعث تضعیف شکل می شود (یعنی مهمترین اجزای فرکانس شکل باید تبدیل کند تا روی علامت گذار ها تأثیر بگذارد).



شکل ۸.۴۹ یک سیستم برچسب گذاری (الف: شکل: الف) رمز گذار ؛ (ب) رمز گشایی





شکل ۸.۵۰ (الف) و (ج) دو نسخه برجسب گذاری از شکل ۸.۹ (الف)؛ (ب) و (د) تفاوتها (در مقیاس شدید) بین نسخه های برجسب گذاری شده و شکل بدون علامت این دو شکل سهم شدت (اگرچه به طور چشمگیری مقیاس بندی شده) از علائم شبه تصادفی آب در شکل اصلی را نشان می دهد.

شکل های ۸.۵۰ (ب) و (د) تبدیلات ی در شدت شکل ایجاد می کنند که ناشی از اعداد شبه تصادفی است که در ضرایب DCT تصاویر برجسب گذاری شده در شکلها تعبیه شده است. ۸.۵۰ (الف) و (ج) قابل مشاهده است. بدیهی است که اعداد شبه تصادفی باید تاثیری (حتی برای مشاهده بسیار کوچک) بر روی تصاویر برجسب گذاری شده داشته باشند. برای نمایش جلوه، تصاویر را در شکل ۱ نشان دهید. ۸.۵۰ (a) و (c) از شکل بدون علامت در شکل ۸.۹ (a) کم شده و شدت آن تا محدوده [۰، ۲۵۵] افزایش یافته است. شکل های ۸.۵۰ (b) و (d) تصاویر حاصله هستند. آنها کمک فضایی دو بعدی از اعداد شبه تصادفی را نشان می دهند. با این حال، زیرا آنها مقیاس بندی شده اند، شما نمی توانید این تصاویر را به سادگی در شکل ۸.۹ (a) به شکل اضافه کنید و تصاویر برجسب گذاری شده را در شکلها دریافت کنید. ۸.۵۰ (الف) و (ج) همانطور که در شکلها مشاهده می شود. ۸.۵۰ (الف) و (ج)، اختلالات شدت حقیقی آنها کوچک تا ناچیز است.

برای تعیین اینکه آیا یک شکل خاص یک کپی از یک شکل برجسب گذاری شده قبلی با علامت آب ۷۱، ۷۲، ...،  $v_k$  و ضرایب  $c_1$  DCT،  $c_2$ ، ...،  $c_k$  است، از روش زیر استفاده می کنیم:

۱. DCT دو بعدی شکل مورد نظر را محاسبه کنید.

۲. ضرایب  $K$  DCT (در موقعیت های مربوط به  $c_1$ ،  $c_2$ ، ...،  $c_k$  مرحله ۲ در روش برجسب گذاری) را استخراج کنید و ضرایب را به عنوان  $\hat{c}_1$ ،  $\hat{c}_2$ ، ...،  $\hat{c}_k$  نشان دهید. اگر شکل مورد نظر شکل برجسب گذاری شده قبلی است (بدون تبدیل)،  $\hat{c}_1$ ،  $\hat{c}_2$ ، ...،  $\hat{c}_k$  در صورتی که کپی اصلاح شده از شکل با علامت گذار باشد (یعنی نوعی حمله را تجربه کرده است)،  $\hat{c}_i = c_i'$  for  $1 \leq i \leq k$ ، تقریبی  $c_i'$  خواهد بود. در غیر این صورت، شکل مورد نظر یک شکل بدون علامت یا شکل با علامت کاملا متفاوت خواهد بود و  $\hat{c}_i$  هیچ شباهتی به  $\hat{c}_i$  اصلی نخواهد داشت.

۳. با استفاده از علامت گذار  $\hat{w}_1, \hat{w}_2, \dots, \hat{w}_k$  محاسبه کنید

$$\hat{w}_i = \frac{\hat{c}_i - c_i}{\alpha c_i} \quad \text{for } 1 \leq i \leq k \quad (8-71)$$

به یاد بیاورید که علامت گذارها توالی اعداد شبه تصادفی هستند.

۴. شباهت  $\hat{w}_1, \hat{w}_2, \dots, \hat{w}_k$  (از مرحله ۳) و  $w_1, w_2, \dots, w_k$  (از مرحله ۳ روش برجسب گذاری آب) را با استفاده از معیارهایی مانند ضریب همبستگی اندازه گیری کنید.

$$\gamma = \frac{\sum_{i=1}^K (\hat{\omega}_i - \bar{\omega})(\omega_i - \bar{\omega})}{\sqrt{\sum_{i=1}^K (\hat{\omega}_i - \bar{\omega})^2 \cdot \sum_{i=1}^K (\omega_i - \bar{\omega})^2}} \quad 1 \leq i \leq K \quad (8-72)$$

جایی که  $\gamma$  و  $\hat{\gamma}$  وسایل دو علامت عنصر  $K$  هستند. (توجه: ضرایب همبستگی در بخش ۱۲.۳ به تفصیل مورد بحث قرار گرفته است.)

۵. شباهت اندازه گیری شده،  $g$  را با یک آستانه از پیش تعیین شده،  $T$  مقایسه کنید و یک تصمیم تشخیص دوتایی بگیرید:

$$D = \begin{cases} 1 & \text{if } \gamma \geq T \\ 0 & \text{otherwise} \end{cases} \quad (8-73)$$

به عبارت دیگر،  $D = 1$  نشان می دهد که علامت گذار  $\gamma_1, \gamma_2, \dots, \gamma_K$  وجود دارد (با توجه به آستانه مشخص شده،  $T$ ).  $D = 0$  نشان می دهد که چنین نبوده است.

با استفاده از این روش، شکل اصلی برچسب گذاری شده در شکل ۸.۵۰ (a)، اندازه گیری شده با خودش، ضریب همبستگی  $0.9999$ ، یعنی  $g = 0.9999$  را به دست می آورد. این یک مسابقه بی تردید است. به شکل مشابه، شکل در شکل ۸.۵۰ (ب)، هنگامی که در مقایسه با شکل در شکل ۸.۵۰ (الف) اندازه گیری می شود،  $g = 0.417$  را نشان می دهد. نمی توان آن را با شکل برچسب گذاری شده در شکل ۸.۵۰ (a) اشتباه گرفت زیرا ضریب همبستگی بسیار پایین است.

برای خاتمه بخش، ما توجه می کنیم که رویکرد برچسب گذاری مبتنی بر DCT مثال قبلی نسبتاً در برابر حملات علامت مقاوم است، تا حدی به این دلیل که یک روش خصوصی یا کلیدهای محدود است. روشهای کلید محدود همیشه نسبت به روشهای کلیدی بدون محدودیت انعطاف پذیرتر هستند. با استفاده از شکل برچسب گذاری شده در شکل ۸.۵۰ (a)، شکل ۸.۵۱ توانایی این روش را برای مقاومت در برابر انواع حملات متداول نشان می دهد. همانطور که در شکل مشاهده می شود، تشخیص علامت در طیف وسیعی از حملات انجام شده بسیار خوب است. ضرایب همبستگی حاصله (در زیر هر شکل در شکل نشان داده شده است) از  $0.3113$  تا  $0.9945$  متغیر است. هنگام قرار گرفتن در معرض کیفیت بالا اما با ضرر (که منجر به خطای  $rms$  شدت می شود) فشرده سازی و رفع فشار JPEG،  $g = 0.9945$  حتی زمانی که فشرده سازی و بازسازی خطای  $rms$  از  $10$  سطح شدت را نشان می دهد،  $g = 0.7395$ ؛ و قابلیت استفاده از این شکل به طور قابل توجهی کاهش یافته است. هموارسازی قابل توجه با فیلترینگ فضایی و افزودن نویز گاوسی ضریب همبستگی را زیر  $0.8230$  کاهش نمی دهد. با این حال، همسان سازی هیستوگرام  $g$  را به  $0.5210$  کاهش می دهد. و چرخش بیشترین تأثیر را دارد. کاهش  $g$  به  $0.3313$ . همه حملات، به استثنای فشرده سازی و بازسازی JPEG در شکل ۸.۵۱ (a)، قابلیت استفاده از شکل برچسب گذاری اولیه را بطور قابل توجهی کاهش داده است.

## خلاصه ، منابع ، و خواندن بیشتر

اهداف اصلی این فصل ارائه مبانی نظری فشرده سازی شکل دیجیتال ، توصیف متداول ترین روش های فشرده سازی و معرفی حوزه مربوط به برچسب گذاری شکل دیجیتال شکل بود. اگرچه سطح ارائه ماهیت مقدماتی دارد ، اما مراجع ورود به مجموعه وسیع ادبیات مربوط به موضوعات مورد بحث را ارائه می دهند. مطابق استانداردهای بین المللی ذکر شده در جداول ۸.۳ تا ۸.۵ ، فشرده سازی نقش کلیدی در ذخیره و تبدیل شکل سند ، اینترنت و توزیع ویدئوهای تجاری (به عنوان مثال ، DVD) دارد. این یکی از معدود حوزه های پردازش شکل است که از نظر تجاری به اندازه کافی مورد توجه قرار گرفته است تا بتواند استانداردهای پذیرفته شده ای را پذیرفته باشد. علامت نگاری شکل به طور فزاینده ای اهمیت پیدا می کند زیرا تصاویر بیشتر به شکل دیجیتال فشرده توزیع می شوند.

مطالب مقدماتی فصل ، که عموماً به بخش ۸.۱ محدود می شود ، برای ایجاد شکل اساسی است و ممکن است در اکثر کتابهای پردازش شکل عمومی که در انتهای فصل ۱ ذکر شده است ، به یک شکل یا دیگری یافت شود. اطلاعات مربوط به سیستم بینایی انسان را ببینید ، Netravali , Limb [۱۹۸۰] ، و همچنین Huang [۱۹۶۶] ، Schreiber and Knapp [۱۹۵۸] ، و منابع ذکر شده در پایان فصل ۲. برای اطلاعات بیشتر در مورد نظریه اطلاعات ، کتاب را ببینید. وب سایت یا آبرامسون [۱۹۶۳] ، بلاهوت [۱۹۸۷] و برگر [۱۹۷۱]. مقاله کلاسیک شانون ، "نظریه ارتباطات ریاضی" [۱۹۴۸] ، پایه و اساس این منطقه را پایه گذاری می کند و یک مرجع عالی دیگر است. معیارهای ذهنی -ذهنی در Frenndall و Behrend [۱۹۶۰] مورد بحث قرار گرفته است. در طول فصل ، انواع مختلفی از استانداردهای فشرده سازی در نمونه ها استفاده شده است. اکثر آنها با استفاده از Adobe Photoshop (با افزودن های فشرده سازی آزاد) و/یا MATLAB ، که در Gonzalez و همکاران توضیح داده شده است ، پیاده سازی شدند. [۲۰۰۴]. استانداردهای فشرده سازی ، به طور معمول ، طولانی و مختلط هستند. ما سعی نکرده ایم هیچ یک از آنها را به طور کامل پوشش دهیم. برای اطلاعات بیشتر در مورد یک استاندارد خاص ، اسناد منتشر شده از سازمان استاندارد مناسب - سازمان استاندارد بین المللی ، کمیسیون بین المللی الکتروتکنیک و/یا اتحادیه بین المللی ارتباطات را ببینید.

تکنیکهای فشرده سازی بدون خطا و فشرده سازی که در بخشهای ۸.۲ تا ۸.۱۱ شرح داده شده است و تکنیکهای برچسب گذاری در بخش ۸.۱۲ ، بیشتر بر اساس مقالات اصلی ذکر شده در متن است. الگوریتم های پوشش داده شده نماینده کار در این زمینه هستند ، اما به هیچ وجه جامع نیستند. مواد مربوط به کدگذاری LZW ریشه در کار Ziv و Lempel دارد [۱۹۷۷ ، ۱۹۷۸]. مطالب مربوط به کدگذاری حسابی در Neal، Witten و Cleary [۱۹۸۷] دنبال می شود. یکی از مهمترین پیاده سازی های کدگذاری حسابی در Pennebaker و همکاران

خلاصه شده است. [۱۹۸۸] برای بحث خوب در مورد برنامه نویسی پیش بینی بدون ضرر ، به آموزش رابانی و جونز [۱۹۹۱] مراجعه کنید. پیش بینی کننده تطبیقی معادله (۸-۵۵) از گراهام است [۱۹۵۸]. برای کسب اطلاعات بیشتر در مورد جبران حرکت ، S. Solari [۱۹۹۷] ، که همچنین مقدمه ای بر استانداردهای کلی فشرده سازی و فشرده سازی ویدئو و میچل و همکاران است را ببینید. [۱۹۹۷]. تکنیک برچسب گذاری مبتنی بر DCT در بخش ۸.۱۲ بر اساس مقاله COX و همکاران است. [۱۹۹۷]. برای اطلاعات بیشتر در مورد برچسب گذاری ، کتابهای COX و همکاران را ببینید. [۲۰۰۱] و پرهی و نیشیتانی [۱۹۹۹]. همچنین به مقاله S. Mohanty [۱۹۹۹] مراجعه کنید.



a b c  
d e f

شکل ۸.۵۱ حمله به شکل برچسب گذاری شده در شکل ۸.۵۰ (الف): (الف) فشرده سازی و رفع فشار JPEG از دست رفته با خطای rms هفت سطح شدت. (ب) فشرده سازی و رفع فشار JPEG از دست رفته با خطای rms ۱۰ سطح شدت (به مصنوع مسدود کننده توجه کنید) ؛ (ج) هموارسازی با فیلترینگ فضایی ؛ (د) افزودن نویز گاوسی ؛ (ه) یکسان سازی هیستوگرام ؛ و (و) چرخش.

هر شکل یک نسخه اصلاح شده از شکل برچسب گذاری شده در شکل ۸.۵۰ (a) است. پس از تبدیل ، آنها علامت های خود را در درجات مختلف حفظ می کنند ، همانطور که توسط ضرایب همبستگی در زیر هر شکل نشان داده شده است.

بسیاری از مقالات نظرسنجی به حوزه فشرده سازی شکل اختصاص داده شده است. قابل توجه Netravali [۱۹۸۰] AK Jain, and Limb [۱۹۸۱] ، یک مسئله ویژه در مورد سیستم های ارتباطی شکل در معاملات IEEE در ارتباطات [۱۹۸۱] ، یک مسئله ویژه در زمینه رمزگذاری گرافیک در مجموعه مقالات IEEE [۱۹۸۰] ، یک مسئله ویژه در مورد سیستم های ارتباط شکل در مجموعه مقالات IEEE [۱۹۸۵] ، یک موضوع ویژه در

فشرده سازی توالی شکل در معاملات IEEE در پردازش شکل [۱۹۹۴]، و یک مسئله ویژه در مورد کمی سازی بردار در معاملات IEEE در پردازش شکل [۱۹۹۶]. علاوه بر این، اکثر مسائل مربوط به تراکنش های IEEE در پردازش شکل، معاملات IEEE روی مدارها و سیستم ها برای فناوری ویدیو و معاملات IEEE در چند رسانه ای شامل مقالاتی در مورد فشرده سازی فیلم و شکل ثابت، جبران حرکت و علامت نگاری می باشد.

## چالش ها و مسائل

راه حل مشکلاتی که با علامت (\*) مشخص شده اند در بسته پشتیبانی دانشجویی DIP $\xi$ E وجود دارد (به وب سایت کتاب مراجعه کنید: [www.ImageProcessingPlace.com](http://www.ImageProcessingPlace.com)).

۸.۱ به موارد زیر پاسخ دهید.

(الف) آیا می توان از روش های کدگذاری با طول متغیر برای فشرده سازی شکل برابر با هیستوگرام با شدت  $n^2$  استفاده کرد؟ توضیح.

(ب) آیا چنین شکل می تواند شامل افزونگی های مکانی یا زمانی باشد که می تواند برای فشرده سازی داده ها مورد سوء استفاده قرار گیرد؟

۸.۲ یکی از انواع کدگذاری طول اجرا شامل می شود

(۱) فقط برنامه نویسی ۰ یا ۱ (نه هر دو) و (۲) اختصاص کد ویژه در ابتدای هر خط برای کاهش اثر خطاهای تبدیل. یک جفت کد ممکن ( $x_k, r_k$ ) است، که در آن  $x_k$  و  $r_k$  به ترتیب نشان دهنده مختصات شروع و طول اجرا هستند. کد (۰، ۰) برای نشان دادن هر خط جدید استفاده می شود.

(الف) یک بیان کلی برای حداکثر میانگین اجرا در هر خط اسکن مورد نیاز برای تضمین فشرده سازی داده ها هنگام کدگذاری طول اجرا یک شکل دوتایی  $2^n \times 2^n$

(ب) حداکثر مقدار مجاز را برای محاسبه کنید  $n = 10$

۸.۳ یک خط ۸ پیکسلی از داده های شدت را در نظر بگیرید، {۱۰۸، ۱۳۹، ۱۳۵، ۲۴۴، ۱۷۲، ۱۷۳، ۵۶، ۹۹}. اگر به طور یکنواخت با دقت ۴ بیتی کوانتیزه شد، خطای rms و نسبت سیگنال به نویز rms را برای داده های کوانتاسیون محاسبه کنید.

۸.۴ \* اگرچه کمی سازی منجر به از دست دادن اطلاعات می شود، اما گاهی اوقات برای چشم نامرئی است. به عنوان مثال، وقتی پیکسل های ۸ بیتی به طور یکنواخت به تعداد پیکسل های کمتری کوانتوم می شوند، اغلب خطوط نادرست رخ می دهد. می توان آن را با استفاده از کوانتیزاسیون مقیاس خاکستری (IGS) کاهش یا حذف کرد. مجموع (ابتدا صفر تنظیم شده است) از مقدار شدت فعلی ۸ بیتی و چهار بیت کمترین مقدار حاصل از مقدار تولید شده قبلی تشکیل می شود. اگر چهار بیت مهمتر از مقدار شدت ۱۱۱۱۲ باشد، به جای آن ۰۰۰۰۲ اضافه می شود. چهار بیت قابل توجه از مجموع حاصل به عنوان مقدار پیکسل کدگذاری شده استفاده می شود.

(الف) کد IGS را برای داده های شدت در مسئله ۸.۳ بسازید.

(ب) خطای rms و نسبت سیگنال به نویز rms برای داده های IGS را محاسبه کنید.

$$A = 10.24 * 24.024$$

شکل ۸ بیتی با ۵.۳ بیت پیکسل آنتروپی [از هیستوگرام آن با استفاده از معادله (۷-۸) محاسبه می شود]. باید توسط هافمن کدگذاری شود.

(الف) حداکثر فشاری که می توان انتظار داشت چقدر است؟

(ب) آیا به دست می آید؟

(ج) اگر سطح بیشتری از فشرده سازی بدون اتلاف مورد نیاز باشد، چه کار دیگری می توان انجام داد؟

۸.۶ \* واحد اطلاعات پایه و معمولاً واحد اطلاعات پایه ۱۰ هارتلی نامیده می شود. فاکتورهای تبدیل مورد نیاز برای ارتباط این واحدها با واحد اطلاعات پایه ۲ (بیت) را محاسبه کنید.

۸.۷ \* ثابت کنید که برای یک منبع حافظه صفر با سیمبولهای  $q$ ، حداکثر مقدار آنتروپی  $\log q$  است که اگر و تنها در صورتی که همه نمادهای منبع به طور یکسان قابل دستیابی باشند بدست می آید. نکته: مقدار کمی  $q$

$$H(z) - \text{را در نظر بگیرید و به نابرابری } \ln x \leq x - 1 \text{ توجه کنید}$$

۸.۸ به موارد زیر پاسخ دهید.

(الف) چند کد منحصر به فرد هافمن برای منبع سه علامت وجود دارد؟

(ب) آنها را بسازید.

۸.۹ شکل ساده ۴، ۸، \*، ۸ بیتی را در نظر بگیرید:

21 21 21 95 169 243 243 243

21 21 21 95 169 243 243 243

21 21 21 95 169 243 243 243

21 21 21 95 169 243 243 243

(الف) آنتروپی شکل را محاسبه کنید.

(ب) شکل را با استفاده از کدگذاری هافمن فشرده کنید.

(ج) فشرده سازی به دست آمده و اثربخشی کدگذاری هافمن را محاسبه کنید.

(د) \* هافمن به جای پیکسل های جداگانه، جفت پیکسل را کدگذاری کنید. به این معنا، تصور کنید که شکل با پسوند دوم منبع حافظه صفر که شکل اصلی را تولید کرده است، تولید شود. وقتی به صورت جفت پیکسل نگاه می شود، آنتروپی شکل چیست؟

(ه) کدگذاری تفاوت بین پیکسل های مجاور را در نظر بگیرید. آنتروپی شکل تفاوت جدید چیست؟ این در مورد فشرده سازی شکل به ما چه می گوید؟

(و) تفاوت های آنتروپی را در (a)، (d) و (e) توضیح دهید.

۸.۱۰ با استفاده از کد هافمن در شکل ۸.۸، رشته کد شده ۰۱۰۱۰۰۰۰۰۱۰۱۰۱۱۱۱۰۱۰۰ را رمزگشایی کنید.  
 ۸.۱۱ کد  $Golomb\ G_3(n)$  را با  $0 \leq n \leq 15$  محاسبه کنید.

۸.۱۲ یک روش کلی برای رمزگشایی کد  $Golomb\ G_m(n)$  بنویسید.

۸.۱۳ چرا نمی توان کد هافمن از اعداد صحیح غیر منفی،  $n \geq 0$  را با تابع احتمال جرم معادله (۸-۱۳) محاسبه کرد؟

۸.۱۴ محاسبه کد نمایی  $Golomb\ G_2(n)$  برای  $0 \leq n \leq 15$ .

یک روش کلی برای رمزگشایی کد  $Golomb$  نمایی  $G_k(n)$  بنویسید.

۸.۱۶ پارامتر کدگذاری مطلوب  $Golomb\ m$  را به عنوان تابعی از  $r$  برای  $0 < \rho < 1$  در معادل (۸-۱۴) ترسیم کنید.

۸.۱۷ با توجه به منبع چهار علامت  $\{d, c, b, a\}$  با احتمال منبع  $\{0.1, 0.4, 0.3, 0.2\}$ ، توالی  $bbadc$  را به صورت حسابی رمزگذاری می کنیم.

۸.۱۸ \* رمزگشایی حسابی برعکس روش رمزگذاری است. با توجه به مدل کدگذاری، پیام ۰.۲۳۳۵۵ را رمزگشایی کنید

Symbol	Probability
a	0.2
e	0.3
i	0.1
o	0.2
u	0.1
!	0.1

۸.۱۹ از الگوریتم کدگذاری  $LZW$  برای کدگذاری رشته ۷ بیتی "ASCII"aaaaaaaaaa استفاده کنید.

۸.۲۰ \* الگوریتمی برای رمزگشایی خروجی کدگذاری شده  $LZW$  مثال ۸.۷ طراحی کنید. از آنجا که دیکشنری مورد استفاده در هنگام رمزگذاری در دسترس نیست، کتاب رمز باید با رمزگشایی خروجی بازتولید شود.

۸.۲۱ دنباله رمزگذاری شده  $BMP$   $\{3, 4, 5, 6, 0, 3, 103, 125, 67, 0, 2, 47\}$  رمزگشایی شود.

۸.۲۲ موارد زیر را انجام دهید:

(الف) کل کد خاکستری ۴ بیتی را بسازید.

(ب) یک روش کلی برای تبدیل یک عدد کد خاکستری به معادل دودویی آن ایجاد کرده و از آن برای رمزگشایی ۰۱۱۱۰۱۰۱۰۱۱۱ استفاده کنید.

۸.۲۳ از الگوریتم فشرده سازی گروه ۴  $CCITT$  برای کدگذاری خط دوم بخش دو خط زیر استفاده کنید:

01100111001111111100001

11111110001110000111111

فرض کنید عنصر مرجع اولیه  $a_0$  در پیکسل اول بخش دوم خط دوم قرار دارد. (توجه: از جدول کد -۲ CCITT D در وب سایت کتاب استفاده کنید).

۸.۲۴ \* موارد زیر را انجام دهید.

(الف) کلیه اعضای طبقه بندی تفاوت ضرایب JPEG DC را فهرست کنید.

(ب) کدهای پیش فرض هافمن خود را با استفاده از جدول کد Huffamn مناسب از وب سایت کتاب محاسبه کنید.

۸.۲۵ چند محاسبات برای یافتن بردار حرکت مطلوب یک ماکرو بلاک با اندازه  $8 * 8$  با استفاده از معیار بهینه سازی MAD، دقت تک پیکسل و حداکثر جابجایی مجاز ۸ پیکسل مورد نیاز است؟ برای چه چیزی می شود  $\frac{1}{4}$  دقت پیکسل؟

۸.۲۶ مزایای استفاده از قاب B برای جبران حرکت چیست؟

۸.۲۷ \* نمودار بلوک رمزگشای ویدئویی جبران شده همراه برای رمزگذار را در شکل ۸.۳۶ ترسیم کنید.

۸.۲۸ شکل که تابع خود همبستگی آن به شکل معادله (۸-۴۸) است. با  $rh=0$  قرار است با استفاده از یک پیش بینی کننده مرتبه دوم کد DPCM داده شود.

(الف) ماتریس R و همبستگی خودکار را تشکیل دهید.

(ب) ضرایب بهینه پیش بینی را پیدا کنید.

(ج) واریانس خطای پیش بینی را که در نتیجه استفاده از ضرایب بهینه حاصل می شود، محاسبه کنید.

۸.۲۹ سطح تصمیم گیری و بازسازی Lloyd-Max را برای  $L = 4$  و تابع چگالی احتمال یکنواخت بدست آورید

$$p(s) = \begin{cases} \frac{1}{2A} & -A \leq s \leq A \\ 0 & \text{otherwise} \end{cases}$$

۸.۳۰ یک رادیولوژیست از یک بیمارستان تحقیقاتی مشهور اخیراً در یک کنفرانس پزشکی شرکت کرد که در آن سیستمی که می توانست تصاویر اشعه ایکس ۱۲۹ بیتی  $4096 * 9096$  را از طریق خطوط تلفن استاندارد T1 (۱.۵۴۴ مگابایت بر ثانیه) منتقل کند به نمایش گذاشته شد. این سیستم تصاویر را در قالب فشرده با استفاده از یک تکنیک پیشرونده منتقل می کند که در آن تقریبی منطقی از اشعه ایکس برای اولین بار در ایستگاه مشاهده بازسازی می شود، سپس به تدریج برای ایجاد یک صفحه نمایش بدون خطا بازسازی می شود. تبدیل داده های مورد نیاز برای ایجاد تقریب اول تقریباً ۵ یا ۶ ثانیه به طول انجامید. گزارشات هر ۵ یا ۶ ثانیه (به طور متوسط) برای ۱ دقیقه بعدی انجام شد، به طوری که اولین و آخرین بازخوانی ها به ترتیب بیشترین و کمترین تأثیر را بر روی اشعه ایکس بازسازی شده داشتند. پزشک تحت تأثیر سیستم قرار گرفت، زیرا می توانست تشخیص خود را با استفاده از تقریب اول اشعه ایکس آغاز کرده و آن را تکمیل کند زیرا بازسازی بدون خطا در اشعه ایکس ایجاد



می شد. پس از بازگشت به دفتر ، او درخواست خرید را به مدیر بیمارستان ارائه کرد. متأسفانه بودجه بیمارستان نسبتاً کمی بود ، که اخیراً با استخدام یک دانشجوی فارغ التحصیل رشته مهندسی برق هزینه شده بود. برای جلب رضایت رادیولوژیست ، مدیر وظیفه طراحی چنین سیستمی را به مهندس جوان داد. (او فکر کرد که طراحی و ساختن یک سیستم مشابه در خانه ارزان تر است. بیمارستان در حال حاضر برخی از عناصر چنین سیستمی را در اختیار دارد ، اما تبدیل داده های خام اشعه ایکس بیش از ۲ دقیقه به طول انجامید.) از مهندس خواسته است تا یک جلسه بلوک اولیه تا جلسه بعد از ظهر کارکنان داشته باشد.

با صرف زمان کمی و تنها یک نسخه از پردازش شکل دیجیتال از دوران مدرسه اخیر ، این مهندس توانست سیستمی را به گونه ای طراحی کند که بتواند نیازهای تبدیل و فشرده سازی را ارضا کند. یک طرح بلوکی مفهومی از چنین سیستمی بسازید و تکنیک های فشرده سازی را که توصیه می کنید مشخص کنید.

۸.۳۱ نشان دهید که تبدیل موجک مبتنی بر بالابر توسط معادله تعریف شده است. (۸-۶۱) معادل پیاده سازی سنتی بانک FWT با استفاده از ضرایب جدول ۷.۱ است. ضرایب فیلتر را از نظر  $a, b, g, d$  و  $K$  تعریف کنید.

۸.۳۲ اندازه گام های کوانتیزاسیون زیر باندها را برای یک شکل رمزگذاری شده JPEG-۲۰۰۰ که در آن از کوانتیزه مشتق شده استفاده شده و ۸ بیت به مانتیسا و ضریب زیر باند LL۲ اختصاص داده شده است ، محاسبه کنید.

۸.۳۳ چگونه می توان یک علامت قابل مشاهده را به یک شکل در حوزه فرکانس اضافه کرد؟

۸.۳۴ \* طراحی یک سیستم برچسب گذاری نامرئی بر اساس تبدیل گسسته فوریه.

۸.۳۵ طراحی یک سیستم برچسب گذاری نامرئی بر اساس تبدیل موجک گسسته.

## فصل نهم : فرآیند شکل شناسی

از نظر شکل و ویژگی ، صورت و اندام ، من مثل برادرم رشد کردم  
این افراد مرا به خاطر او و هر یک را برای یکدیگر می گرفتند.  
هنری سامبروک لی ، سرودهای کوکاین ، دوقلوها

### پیش نمایش

واژه مورفولوژی (شکل شناسی) معمولاً نشان دهنده شاخه ای از زیست شناسی است که به شکل و ساختار  
حیوانات و گیاهان می پردازد. ما از همین کلمه در زمینه شکل شناسی ریاضی به عنوان ابزاری برای استخراج  
اجزای شکل استفاده می کنیم که در نمایش و توصیف شکل ناحیه مانند مرزها ، طرح ها و بدنه محدب مفید  
هستند. ما همچنین به تکنیک های شکل شناسی برای قبل یا بعد از پردازش مانند فیلتر شکل شناسی ، نازک  
شدن و هرس علاقه داریم.

در بخشهای بعدی ، ما تعدادی از مفاهیم اساسی در شکل شناسی ریاضی را توسعه داده و نحوه کاربرد آنها در  
پردازش شکل را توضیح خواهیم داد. مطالب موجود در این فصل از روشهایی که ورودی و خروجی آنها شکل  
است ، به روشهایی که خروجی آنها ویژگیهای شکل است ، برای کارهایی مانند استخراج و شرح شی شروع می  
شود. شکل شناسی یکی از چندین ابزار توسعه یافته در بقیه کتاب است- مانند تقسیم بندی ، استخراج ویژگی  
ها و تشخیص اشیاء- که پایه و اساس تکنیک های استخراج "معنا" از یک شکل را تشکیل می دهد. مطالب  
بخشهای بعدی این فصل به روشهای پردازش تصاویر باینری و مقیاس خاکستری می پردازد.

### پس از اتمام این فصل ، خوانندگان باید:

درک مفاهیم اساسی شکل شناسی ریاضی و نحوه استفاده از آنها در پردازش شکل دیجیتال.

با ابزارهای مورد استفاده برای شکل شناسی شکل دوتایی، از جمله سایش، اتساع، باز شدن، بسته شدن و نحوه ترکیب آنها برای تولید ابزارهای مختلط آشنا باشید.

قادر به توسعه الگوریتم های مبتنی بر شکل شناسی شکل دوگانه برای انجام وظایف باشید مانند هموارسازی شکل شناسی، حذف لبه، استخراج اجزای متصل و طرح سازی.

با چگونگی گسترش شکل شناسی شکل باینری به تصاویر در مقیاس خاکستری آشنا باشید.

قادر به توسعه الگوریتم هایی برای پردازش شکل در مقیاس خاکستری برای کارهایی مانند تقسیم بندی بافت، گرانولومتری، محاسبه شیب شکل در مقیاس خاکستری و سایر موارد باشید.

قبل از ادامه، بررسی بحث در بخش ۲.۴ در مورد نمایش تصاویر، بحث در مورد اتصال در بخش ۲.۵ و بحث در مورد مجموعه ها در بخش ۲.۶ مفید خواهد بود.

## ۹.۱. مقدماتی

زبان شکل شناسی ریاضی نظریه مجموعه است. به این ترتیب، شکل شناسی یک رویکرد یکپارچه و قدرتمند برای مشکلات متعدد پردازش شکل ارائه می دهد. هنگام کار با تصاویر، مجموعه های شکل شناسی ریاضی اشیاء موجود در آن تصاویر را نشان می دهند. در تصاویر دوتایی، مجموعه های مورد نظر اعضای فضای صحیح دو بعدی  $Z^2$  هستند، جایی که هر عنصر یک مجموعه یک تاپل (بردار دو بعدی) است که مختصات آن مختصات یک پیکسل یک جسم (معمولاً پیش زمینه) در شکل است. به تصاویر دیجیتالی مقیاس خاکستری را می توان به عنوان مجموعه هایی که اجزای آنها در  $Z^3$  هستند نمایش داد. در این حالت، دو جزء از هر عنصر مجموعه به مختصات یک پیکسل اشاره می کند و سومی مربوط به مقدار شدت گسسته آن است. مجموعه ها در فضاهای با ابعاد بالاتر می توانند شامل ویژگی های دیگر شکل مانند رنگ و اجزای متغیر در زمان باشند.

عملیات شکل شناسی از نظر مجموعه تعریف شده است. در پردازش شکل، ما از شکل شناسی با دو نوع مجموعه پیکسل استفاده می کنیم: اشیاء و عناصر ساختار (SE). به طور معمول، اشیاء به عنوان مجموعه ای از پیکسل های پیش زمینه تعریف می شوند. عناصر ساختار را می توان از نظر پیکسل های پیش زمینه و پس زمینه مشخص کرد. علاوه بر این، عناصر ساختاری گاهی اوقات حاوی عناصر به اصطلاح "اهمیت ندهید" هستند که با  $\emptyset$  نشان داده می شود، که نشان می دهد ارزش آن عنصر خاص در SE مهم نیست. در این معنا، ارزش را می توان نادیده گرفت، یا می توان آن را در ارزیابی یک عبارت به مقدار مطلوب رساند. به عنوان مثال، ممکن است مقدار پیکسل در یک شکل در برنامه هایی که تطابق مقدار هدف است، به خود بگیرد.

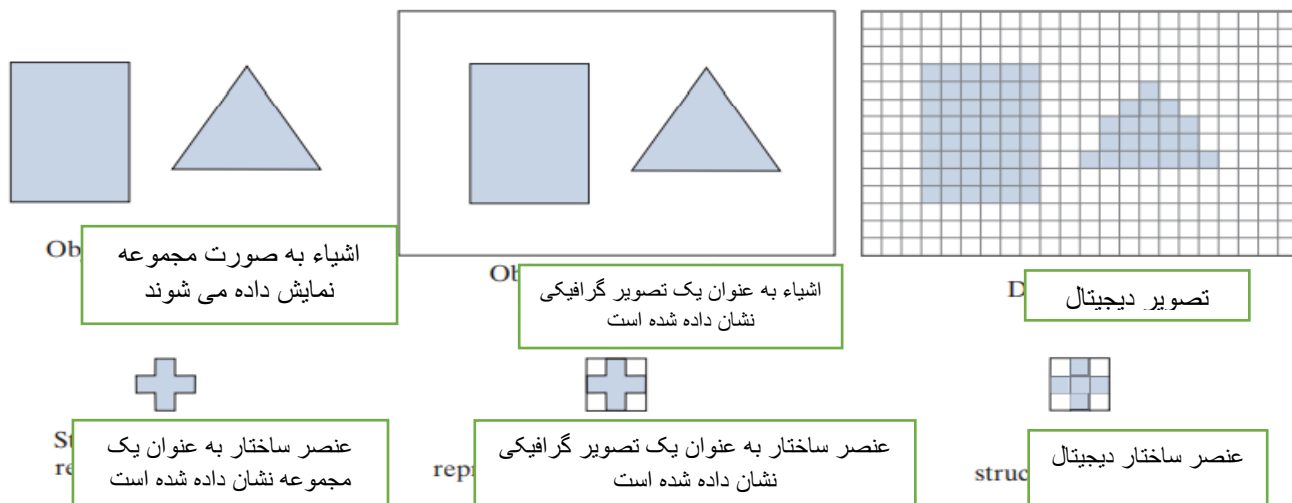
از آنجا که تصاویری که ما با آنها کار می کنیم آرایه های مستطیلی هستند و مجموعه ها به طور کلی دارای شکل دلخواه هستند، کاربردهای شکل شناسی در پردازش شکل مستلزم این است که مجموعه ها در آرایه های مستطیلی تعبیه شوند. در شکل گیری چنین آرایه هایی، ما به تمام پیکسل هایی که اعضای مجموعه های شیء

نیستند ، مقدار زمینه اختصاص می دهیم. سطر بالا در شکل ۹.۱ نمونه ای را نشان می دهد. در سمت چپ مجموعه هایی با فرمت گرافیکی وجود دارد که عادت کرده اید در تصاویر کتاب مشاهده کنید. در مرکز ، مجموعه ها در یک زمینه مستطیلی (سفید) برای ایجاد یک شکل<sup>۴۳</sup> گرافیکی جاسازی شده اند.

عناصر ساختار به همان شیوه تعریف شده اند ، و ردیف دوم در شکل ۹.۱ نمونه ای را نشان می دهد. تفاوت مهمی بین نحوه نمایش تصاویر دیجیتال و عناصر ساختار دیجیتال وجود دارد. در بالا سمت راست توجه کنید که حاشیه ای از پیکسل های پس زمینه در اطراف اجسام وجود دارد ، در حالی که در SE وجود ندارد. همانطور که به زودی خواهید آموخت ، عناصر ساختاردهی به شکلی شبیه به هسته های کانولوشن فضایی مورد استفاده قرار می گیرند (شکل ۳.۲۸ را ببینید) ، و حاشیه شکل که توضیح داده شد مشابه لایه ای است که در بخش ۳.۴ و ۳.۵ مورد بحث قرار دادیم. عملیات از نظر شکل شناسی متفاوت است ، اما عملیات روکش و کشویی همانند حرکت پیچشی است.

علاوه بر تعاریف مجموعه ارائه شده در بخش ۲.۶ ، مفهوم بازتاب و تبدیل مجموعه در شکل شناسی در ارتباط با عناصر ساختاربخش بسیار مورد استفاده قرار می گیرد. بازتاب مجموعه (عناصر ساختاردهنده) B در مورد منشأ آن ، که با  $\hat{B}$  مشخص می شود ، به صورت زیر تعریف می شود:

$$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\} \quad (9-1)$$



شکل ۹.۱ ردیف بالا، چپ: اشیاء به صورت مجموعه های گرافیکی نشان داده می شوند. مرکز: اشیاء تعبیه شده در پس زمینه برای ایجاد یک شکل گرافیکی. راست: شی و پس زمینه دیجیتالی می شوند تا یک شکل دیجیتالی

<sup>۴۳</sup> مجموعه ها به صورت نقشه های اجسام (مانند مربع و مثلث) با شکل دلخواه نشان داده می شوند. یک تصویر گرافیکی شامل مجموعه هایی است که برای ایجاد یک آرایه مستطیلی در پس زمینه جاسازی شده اند. وقتی قصد داریم یک نقاشی به عنوان یک تصویر دیجیتالی (یا عنصر ساختاردهنده) تفسیر شود ، یک شبکه را در تصاویر گنجانده ایم که در غیر این صورت ممکن است مبهم باشد. اشیاء در تمام نقاشی ها سایه دار هستند و پس زمینه به رنگ سفید نشان داده شده است. هنگام کار با تصاویر باینری واقعی ، می گوئیم که اشیاء پیکسل های پیش زمینه هستند. همه پیکسل های دیگر پس زمینه هستند.

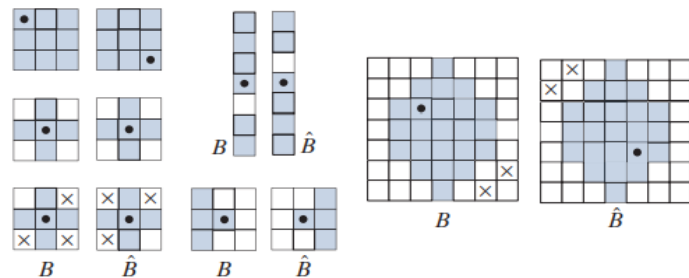
ایجاد کنند (به شبکه توجه کنید). ردیف دوم: نمونه ای از عنصر ساختار بندی که به صورت مجموعه ، شکل گرافیکی و در نهایت به عنوان SE دیجیتال نشان داده شده است.

یعنی اگر B مجموعه ای از نقاط در  $D-2$  باشد ،  $\hat{B}$  مجموعه نقاطی در B است که مختصات  $(y, x)$  آنها با  $(-y, -x)$  جایگزین شده است. شکل ۹.۲ چندین نمونه از مجموعه های دیجیتال (ساختار عناصر) و بازتاب آنها را نشان می دهد. نقطه منشا SE را نشان می دهد. توجه داشته باشید که بازتاب صرفاً شامل چرخاندن یک SE به میزان  $180^\circ$  درجه در مورد مبدأ آن است و همه عناصر ، از جمله عناصر پس زمینه و عناصر بی اهمیت ، چرخانده می شوند. تبدیل مجموعه B بر اساس نقطه  $z = (z_1, z_2)$  ، با علامت  $(B)_z$  ، به صورت

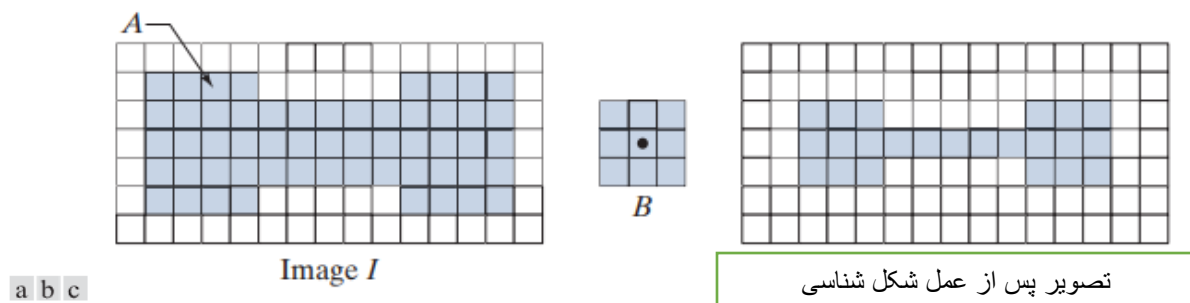
$$(B)_z = \{c \mid c = b + z, \text{ for } b \in B\} \quad (9-2)$$

بازتاب همان عملی است که ما قبل از تبدیل فضایی با هسته ها انجام دادیم ، همانطور که در بخش ۳.۴ توضیح داده شده است.

یعنی اگر B مجموعه ای از پیکسل ها در  $D-2$  باشد ، پس  $(B)_z$  مجموعه ای از پیکسل ها در B است که مختصات  $(y, x)$  آنها با  $(x + z_1, y + z_2)$  جایگزین شده است. این ساختار برای تبدیل دیر (اسلاید) یک عنصر ساختاردهی بر روی یک شکل ، و هر مکان مجموعه ای را انجام می دهد



شکل ۹.۲ ساختار بندی عناصر و بازتاب آنها در مورد مبدأ (عناصر عنصر S عناصر بی اهمیتی هستند و نقطه ها منشاء را نشان می دهند). انعکاس عبارت است از چرخش  $180^\circ$  درجه SE در مورد منشاء آن.



شکل ۹.۳ (الف) یک شکل دودویی حاوی یک جسم (مجموعه) ، الف (ب) یک عنصر سازه ، B. (ج) شکل حاصل از یک عملیات شکل شناسی (متن را ببینید).

همانطور که در شکل ۳.۲۸ برای همبستگی و تبدیل توضیح دادیم، بین عنصر ساختاردهی و ناحیه شکل مستقیماً زیر آن عمل می‌شود. هر دو بازتاب و تبدیل با توجه به منشاء B تعریف می‌شوند.

به عنوان مقدمه ای بر نحوه انجام عملیات شکل شناسی بین تصاویر و عناصر ساختاردهی، شکل ۹.۳ را در نظر بگیرید، که یک شکل دوتایی ساده را نشان می‌دهد، A، متشکل از یک شیء (مجموعه) A، نشان داده شده با سایه، و  $3 * 3$  که عناصر آن همه ۱ هستند (پیکسل های پیش زمینه). پیکسل های پس زمینه (۰) به رنگ سفید نشان داده شده است. ما علاقمند به انجام عملیات شکل شناسی زیر هستیم: (۱) ایجاد یک شکل جدید، به اندازه من، که در ابتدا فقط از مقادیر پس زمینه تشکیل شده است، (۲) ترانس (اسلاید) B روی شکل A و (۳) در هر افزایش تبدیل، اگر B به طور کامل در A موجود باشد، محل منشاء B را به عنوان پیکسل پیش زمینه در شکل جدید برچسب گذاری کنید. در غیر این صورت، آن را به عنوان یک پس زمینه بگذارید. شکل ۹.۳ (ج) نتیجه ای است که مبدأ B از همه عناصر A بازدید کرده است. ما می‌بینیم که وقتی مبدأ B روی یک عنصر حاشیه ای A باشد، قسمتی از B در A متوقف می‌شود، بنابراین حذف می‌شود محل منشاء B به عنوان نقطه پیش زمینه احتمالی شکل جدید. نتیجه خالص این است که مرز مجموعه A فرسوده می‌شود، همانطور که در شکل ۹.۳ (e) نشان داده شده است. به دلیل روشی که ما عملیات را تعریف کردیم، حداکثر گشت مورد نیاز برای B در A زمانی است که مبدأ B (که در مرکز آن قرار دارد) در A وجود داشته باشد. با اندازه  $3 * 3$ ، باریک ترین پد زمینه عرض مورد نیاز ما یک پیکسل بود، همانطور که در شکل ۹.۳ (a) نشان داده شده است. با استفاده از کوچکترین حاشیه مورد نیاز برای عملیات، نقشه‌ها را کوچکتر نگه می‌داریم. در عمل، ما عرض پد را بر اساس حداکثر ابعاد عناصر ساختار استفاده شده، صرف نظر از عملیات انجام شده، مشخص می‌کنیم. وقتی از اصطلاحاتی مانند "عنصر ساختار B در مجموعه A استفاده می‌شود" استفاده می‌کنیم، منظور ما این است که عناصر پیش زمینه B فقط عناصر A را با هم تداخل دارند. عناصر مراقبت همچنین، ما از مجموعه A برای نشان دادن تمام پیکسل های پیش زمینه A استفاده می‌کنیم. این عناصر پیش زمینه می‌توانند مانند شکل ۹.۳ یک شیء واحد باشند یا می‌توانند زیر مجموعه های جدا از عناصر پیش زمینه را نشان دهند، مانند ردیف اول شکل ۹.۱. ما در مورد تصاویر دوتایی و ساختار عناصر بخش ۹.۲ تا ۹.۷ بحث خواهیم کرد. سپس، در بخش ۹.۸، ایده های دوتایی را به تصاویر مقیاس خاکستری و ساختار عناصر بسط می‌دهیم.

دلیل اینکه ما عموماً مرز لایه را با ابعاد B مشخص می‌کنیم این است که برخی از عملیات شکل شناسی برای یک عنصر ساختار کامل تعریف شده است و نمی‌توان با توجه به محل منشاء آن را تفسیر کرد.

## ۹.۲ سایش و اتساع

ما بحث شکل شناسی را با مطالعه دو عملیات سایش و اتساع آغاز می کنیم. این عملیات برای پردازش شکل شناسی اساسی است. در واقع، بسیاری از الگوریتم های شکل شناسی مورد بحث در این فصل بر اساس این دو عملیات اولیه است.

### سایش

عبارات شکل شناسی بر اساس ساختار عناصر و مجموعه ای،  $A$ ، پیکسل های پیش زمینه، یا از نظر ساختار عناصر و یک شکل  $I$ ، حاوی  $A$  نوشته می شوند. ما ابتدا رویکرد قبلی را در نظر می گیریم. با  $A$  و  $B$  به عنوان مجموعه در  $Z^2$ ، سایش  $A$  توسط  $B$ ، نشان دهنده  $A \ominus B$  است | است  $B$ ، به صورت تعریف شده است

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \quad (9-3)$$

به یاد داشته باشید، مجموعه  $A$  می تواند نشان دهنده (اتحاد) چندین مجموعه جدا از هم پیکسل های پیش زمینه (یعنی اشیاء) باشد.

جایی که  $A$  مجموعه ای از پیکسل های پیش زمینه است،  $B$  یک عنصر ساختار دهنده است و  $Z$  ها مقادیر پیش زمینه هستند (۱). به عبارت دیگر، این معادله نشان می دهد که سایش  $A$  توسط  $B$  مجموعه ای از تمام نقاط  $Z$  است به طوری که  $B$ ، تبدیل  $Z$ ، در  $A$  وجود دارد (به یاد داشته باشید، جابجایی با توجه به مبدا  $B$  تعریف می شود). معادله (۳-۹) فرمول حاصل از پیکسل های پیش زمینه شکل در شکل ۹.۳ (ج). همانطور که اشاره شد، ما با مجموعه ای از پیکسل های پیش زمینه که در مجموعه ای از پیکسل های پشت زمینه تعبیه شده اند کار می کنیم تا یک شکل کامل ایجاد کنیم. ما می توانیم این واقعیت را با نوشتن معادله (۳-۹) صریح بیان کنیم. به عنوان

$$I \ominus B = \{z \mid (B)_z \subseteq A \text{ and } A \subseteq I\} \cup \{A^c \mid A^c \subseteq I\} \quad (9-4)$$

جایی که من یک آرایه مستطیلی از پیکسل های پیش زمینه و پس زمینه هستیم. محتویات مهاربندهای اول همان معادله (۳-۹)، را بیان می کند. با توضیح اضافه شده که  $A$  زیرمجموعه ای است (یعنی در آن موجود است)  $A$ . پیوند با عملیات داخل مجموعه دوم براکت ها پیکسل هایی را که در زیرمجموعه  $A$  نیستند، "اضافه" می کند (یعنی،  $A^c$ ، که مجموعه ای از پیکسل های پشت زمینه است) نتیجه اولین مهاربندها است، همچنین لازم است پیکسل های پس زمینه بخشی از مستطیل تعریف شده توسط  $A$  باشند. به عبارت دیگر، تمام این معادله می گوید سایش  $A$  توسط  $B$  مجموعه ای از تمام نقاط  $Z$ ، به طوری که  $B$ ، با  $Z$  تبدیل شده است، در آن وجود دارد. الف) معادله همچنین صریح می کند که  $A$  در  $I$  موجود است، که نتیجه در مجموعه ای از پیکسل های پس زمینه جاسازی شده است و کل فرایند با اندازه  $I$  یکسان است. البته، ما از دست و پا گیر استفاده نمی کنیم نماد معادله

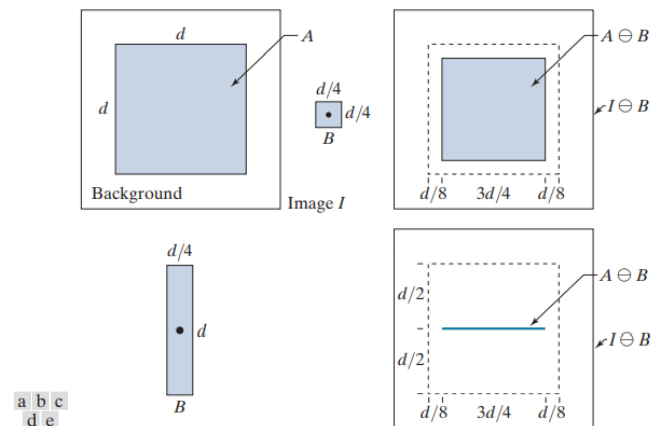
(۴-۹) ، که ما فقط برای تأکید بر یک نکته مهم نشان می دهیم. در عوض ، ما از علامت  $A$  | استفاده می کنیم  $B$  وقتی یک عملیات شکل شناسی فقط از عناصر پیش زمینه استفاده می کند ، و  $B$  | هنگامی که عملیات از عناصر پیش زمینه و پس زمینه استفاده می کند. این تمایز ممکن است بی اهمیت به نظر برسد ، اما فرض کنید ما می خواهیم سایش را با معادله (۳-۹) ، انجام دهیم. با استفاده از عناصر پیش زمینه عنصر ساختار در آخرین ستون در شکل ۹.۲. این عنصر ساختاریبخش نیز دارای عناصر پس زمینه است ، اما معادله (۳-۹) فرض می کند که  $B$  فقط دارای عناصر پیش زمینه است. در حقیقت ، سایش فقط برای عملیات بین عناصر پیش زمینه تعریف می شود ، بنابراین  $B$  | بدون "توضیح" تعبیه شده در معادله بی معنی خواهد بود. (۴-۹). برای جلوگیری از سردرگمی ، از  $A$  در عبارات شکل شناسی هنگامی که عملیات فقط عناصر پیش زمینه را شامل می شود ، و  $A$  وقتی عملیات شامل عناصر پس زمینه و/یا "اهمیتی نمی دهد" از  $A$  استفاده می کنیم. ما همچنین از استفاده از نمادهای شکل شناسی استاندارد مانند | هنگام کار با  $SE$  های "مخلوط". به عنوان مثال ، بعداً در معادله (۹-۱۷) ما از علامت  $\textcircled{S}$  در عبارت  $B = \{z \mid P\{B\}z \wedge A\}$  استفاده می کنیم که دارای همان فرم معادله (۳-۹) ، اما در عوض شامل کل شکل و مقدار مختلط می شود.  $SE$  در آخرین ستون شکل ۹.۲. همانطور که مشاهده خواهید کرد ، استفاده از  $SE$  با مقادیر مخلوط قدرت قابل توجهی را به عملیات شکل شناسی می افزاید. بازگشت به بحث ما در مورد معادله (۳-۹) ، زیرا این جمله که  $B$  باید در  $A$  موجود باشد معادل  $B$  است که هیچ عنصر مشترکی را با پس زمینه (به عنوان مثال ، مجموعه مکمل  $A$ ) به اشتراک نمی گذارد ، بنابراین می توان سایش را معادله

$$A \ominus B = \{z \mid (B)_z \cap A^c = \emptyset\} \quad (9-5)$$

جایی که ، همانطور که در بخش ۲.۶ تعریف شده است ، مجموعه تهی است. شکل ۹.۴ نمونه ای از سایش را نشان می دهد. عناصر مجموعه  $A$  (سایه دار) پیکسل های پیش زمینه شکل  $A$  هستند و مانند گذشته ، پس زمینه به رنگ سفید نشان داده شده است. مرز جامد درون مرز خط کشی شده در شکل ۹.۴ (ج) محدودیتی است که خارج از آن جایجایی بیشتر منشاء  $B$  باعث می شود برخی عناصر عنصر سازه به طور کامل در  $A$  متوقف شوند. بنابراین ، مکان نقاط (مکانهای منشاء  $B$ ) در داخل (و شامل) این مرز عناصر پیش زمینه سایش  $A$  توسط  $B$ . را تشکیل می دهد. ما سایش حاصله را در شکل ۹.۴ (C) سایه می زنیم ، و زمینه را سفید نشان می دهیم. سایش مجموعه ای از مقادیر  $Z$  است که معادلهای راضی می کند. (۳-۹) یا (۵-۹). مرز  $A$  در شکل نشان داده شده است. ۹.۴ (ج) و (ه) به عنوان مرجع ؛ بخشی از سایش نیست شکل ۹.۴ (d) یک عنصر ساختار کشیده و شکل ۹.۴ (e) سایش  $A$  توسط این عنصر را نشان می دهد. توجه داشته باشید که شیء اصلی در یک خط فرسوده شده است. همانطور که می بینید ، نتیجه سایش توسط شکل عنصر ساختار کنترل می



شود. در هر دو مورد ، فرض بر این است که شکل برای جا به جایی همه گشت های B پد شده است و نتیجه همان اندازه شکل اصلی بریده شده است ، درست مانند آنچه در فصل ۳ با پردازش فضایی انجام دادیم. معادلات (۹-۳) و (۹-۵) تنها تعاریف سایش نیستند (برای دو تعریف اضافی و معادل به مسائل ۹.۱۲ و ۹.۱۳ مراجعه کنید). با این حال ، معادلات قبلی این مزیت را دارند که وقتی عنصر ساختار B به عنوان یک هسته فضایی در حال حرکت است که روی یک مجموعه حرکت می کند ، مانند پیچیدگی ، بصری تر باشند.



شکل ۹.۴ (الف) شکل A ، متشکل از یک مجموعه (شیء) A ، و پس زمینه. (ب) مربع SE ، B (نقطه مبدا است). (ج) سایش A توسط B (در شکل بدست آمده در سایه نشان داده شده است). (د) SE طولانی. (ه) سایش A توسط B. (سایش یک خط است). مرز نقطه ای در (C) و (e) مرز A است که برای مرجع نشان داده شده است.

#### مثال ۹.۱: استفاده از سایش برای حذف اجزای شکل .

شکل ۹.۵ (الف) یک شکل دودویی است که یک ماسک سیم ساده را نشان می دهد. همانطور که قبلاً ذکر شد ، ما به طور کلی پیکسل های پیش زمینه را در تصاویر دوتایی سفید و پس زمینه را سیاه نشان می دهیم. فرض کنید ما می خواهیم خطوط متصل کننده منطقه مرکزی به پدهای حاشیه ای در شکل ۹.۵ (الف) را حذف کنیم. همانطور که در شکل ۹.۵ (ب) نشان داده شده است ، سایش شکل (یعنی سایش پیکسل های پیش زمینه شکل ) با یک عنصر ساختار مربع اندازه  $11 * 11$  که اجزای آن همه ۱ هستند حذف شده است.

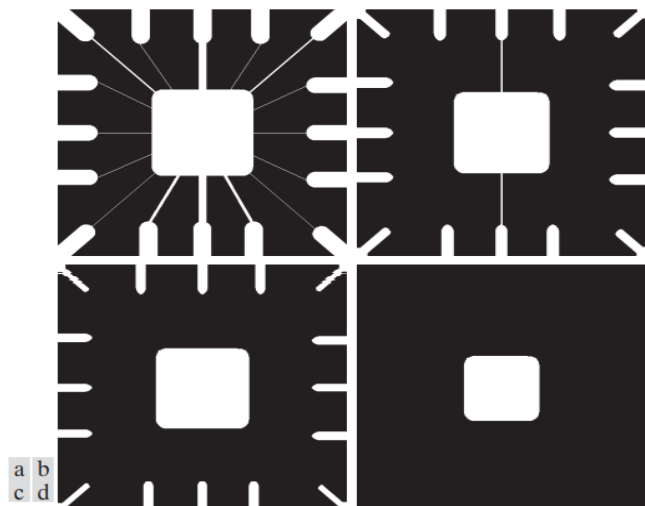
دلیل اینکه دو خط عمودی در مرکز نازک شده اند اما به طور کامل حذف نشده اند این است که عرض آنها بیشتر از  $11$  پیکسل است. تبدیل اندازه SE به  $15 * 15$  عنصر و سایش مجدد شکل اصلی ، همانطور که شکل ۹.۵ (ج) نشان می دهد ، تمام خطوط اتصال را حذف کرد. یک روش جایگزین این بود که شکل را در شکل ۹.۵ (ب) دوباره با استفاده از همان  $11 * 11$  SE یا کوچکتر سایش دهد. افزایش بیشتر عنصر ساختار حتی اجزای بزرگتر را حذف می کند. به عنوان مثال ، همانطور که در شکل ۹.۵ (d) نشان داده شده است ، می توان خطوط اتصال و پدهای حاشیه را با یک عنصر ساختاردهی با اندازه  $45 * 45$  عنصر بر روی شکل اصلی حذف کرد.

از این مثال می بینیم که سایش اجسام را در یک شکل دوتایی کوچک یا نازک می کند. در واقع ، ما می توانیم سایش را به عنوان یک عمل فیلتر شکل شناسی در نظر بگیریم که در آن جزئیات شکل کوچکتر از عنصر ساختار از شکل فیلتر شده (حذف می شوند). در شکل ۹.۵ ، سایش تابع "فیلتر خط" را انجام می دهد. ما به مفهوم فیلترهای شکل شناسی در بخشهای ۹.۴ و ۹.۸ برمی گردیم.

### اتساع

با  $A$  و  $B$  به عنوان مجموعه در  $Z^2$  ، اتساع  $A$  توسط  $B$  ، نشان داده شده به عنوان  $A \oplus B$  ، به صورت زیر تعریف می شود:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \quad (9-6)$$



شکل ۹.۵ استفاده از سایش برای حذف اجزای شکل (الف)  $486 * 486$  شکل دوتایی از ماسک سیم پیوندی که در آن پیکسل های پیش زمین با رنگ سفید نشان داده شده است. (ب) - (د) شکل با استفاده از عناصر ساختار مربع اندازه  $11 * 11$  ،  $15 * 15$  ، فرسوده شده است و  $45 * 45$  عنصر ، به ترتیب ، همه ۱ ارزش دارند. این معادله بر اساس بازتاب  $B$  در مورد منشأ آن و تبدیل بازتاب با  $Z$  ، مانند سایش است. سپس اتساع  $A$  توسط  $B$  مجموعه ای از همه جابجایی ها ،  $Z$  ، از جمله است. عناصر پیش زمینه  $\hat{B}$  حداقل با یک عنصر  $A$  همپوشانی دارند (به یاد داشته باشید ،  $Z$  جابجایی منبع  $\hat{B}$  است). بر اساس این تفسیر ، معادله (۹-۶) را می توان معادل به صورت زیر نوشت

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \quad (9-7)$$

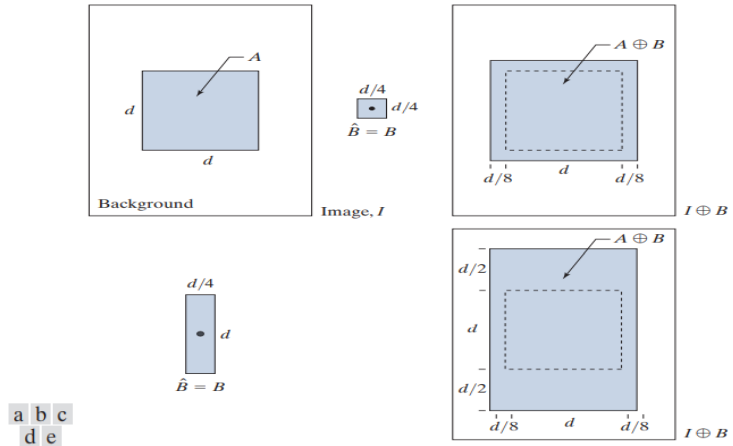
معادلات (۹-۶) و (۹-۷) تنها تعریفی از اتساعی نیستند که در حال حاضر استفاده می شوند (برای دو تعریف متفاوت و در عین حال معادل به مسائل ۹.۱۴ و ۹.۱۵ مراجعه کنید). همانطور که در مورد سایش ، تعاریف قبلی این مزیت را دارند که وقتی ساختار عنصر  $B$  به عنوان یک هسته پیچشی در نظر گرفته می شود ، بصری تر باشد.

همانطور که قبلاً نیز اشاره شد، فرایند اصلی پر کردن (چرخاندن) B در مورد منشاء آن و سپس جابجایی پی در پی آن به طوری که روی مجموعه A لغزیده باشد، مشابه پیچیدگی فضایی است. با این حال، به خاطر داشته باشید که اتساع بر اساس عملیات مجموعه است و بنابراین یک عملیات غیر خطی است، در حالی که کانولوشن مجموع محصولات است، که یک عملیات خطی است.

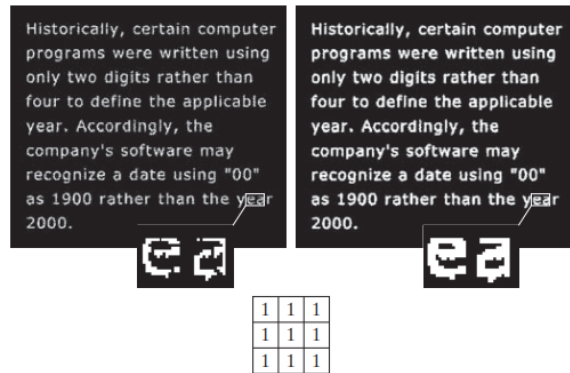
برخلاف سایش، که یک عمل کوچک کننده یا نازک کننده است، اتساع اجسام را در یک شکل دوتایی "رشد" یا "ضخیم" می کند. نحوه و میزان این ضخیم شدن توسط شکل و اندازه عنصر ساختار استفاده شده کنترل می شود. شکل ۹.۶ (الف) همان شیء مورد استفاده در شکل ۹.۴ را نشان می دهد (مساحت پس زمینه بزرگتر است تا محل قرارگیری)، و شکل ۹.۶ (ب) یک عنصر ساختاری را نشان می دهد (در این مورد  $B^{\wedge} = B$  چون SE منشاء آن متقارن است). خط تیره در شکل ۹.۶ (ج) مرز شیء اصلی را برای مرجع نشان می دهد، و خط جامد محدودیتی را نشان می دهد که فراتر از آن هرگونه جابجایی بیشتر مبدأ  $B^{\wedge}$  توسط Z باعث تهی شدن تقاطع  $B^{\wedge}$  و A می شود. به بنابراین، تمام نقاط داخل و داخل این مرز، بسط A توسط B را تشکیل می دهند. شکل ۹.۶ (د) یک عنصر ساختاری را نشان می دهد که برای دستیابی به اتساع بیشتر به صورت عمودی نسبت به افقی طراحی شده است، و شکل ۹.۶ (e) اتساع حاصل از این را نشان می دهد. عنصر

مثال ۹.۲: استفاده از اتساع برای ترمیم کاراکترهای شکسته در یک شکل.

یکی از ساده ترین کاربردهای اتساع برای پر کردن شکاف ها است. شکل ۹.۷ (الف) همان شکل را با کاراکترهای شکسته نشان می دهد که در شکل ۴.۴۸ در ارتباط با فیلترینگ پایین گذر مطالعه کرده ایم. حداکثر طول وقفه ها دو پیکسل است. شکل ۹.۷ (ب) یک عنصر ساختاری را نشان می دهد که می تواند برای ترمیم شکاف ها استفاده شود. همانطور که قبلاً ذکر شد، ما از سفید (۱) برای نشان دادن پیش زمینه و سیاه (۰) برای پس زمینه هنگام کار با تصاویر استفاده می کنیم. شکل ۹.۷ (ج) نتیجه اتساع شکل اصلی با عنصر ساختار را نشان می دهد. فاصله ها پر شد. یکی از مزایای مهم روش شکل شناسی نسبت به روش فیلترینگ پایین گذر که برای پر کردن شکاف در شکل ۴.۴۸ استفاده کردیم این است که روش شکل شناسی مستقیماً به یک شکل دوتایی منجر شد. از طرف دیگر، فیلترینگ پایین گذر با یک شکل دوتایی شروع شد و یک شکل در مقیاس خاکستری ایجاد کرد که برای تبدیل مجدد آن به حالت دوتایی نیاز به آستانه دارد (در فصل ۱۰ در مورد آستانه بحث خواهیم کرد). توجه داشته باشید که مجموعه A در این برنامه شامل اجسام متعدد جدا شده از پیکسل های پیش زمینه است.



شکل ۹.۶ (الف) شکل  $A$ ، متشکل از مجموعه (شی)  $A$  و پس زمینه.  $d/4$  (ب) مربع SE (نقطه مبدا است). (ج) اتساع  $A$  توسط  $B$  (با سایه نشان داده شده است). (د) SE طولانی. (ه) اتساع  $A$  توسط این عنصر. خط نقطه در (C) و (e) مرز  $A$  است که برای مرجع نشان داده شده است.



شکل ۹.۷ (الف) متن با وضوح پایین که کاراکترهای شکسته را نشان می دهد (به نمای بزرگنمایی شده مراجعه کنید). (ب) عنصر ساختار. (ج) اتساع (الف) توسط (ب). بخش های شکسته به هم پیوستند.

### دوگانگی

سایش و اتساع نسبت به تکمیل و بازتاب مجموعه های دوگانه یکدیگر هستند. به این معنا که،

$$(A \oplus B)^c = A^c \oplus \hat{B} \quad (9-8)$$

$$(A \oplus B)^c = A^c \ominus \hat{B} \quad (9-9)$$

معادله (۹-۸) نشان می دهد که سایش  $A$  توسط  $B$  مکمل اتساع  $AC$  توسط  $\hat{B}$  است و بالعکس. ویژگی دوگانگی زمانی مفید است که مقادیر عنصر ساختاربخش نسبت به منشاء آن متقارن باشند (همانطور که در اغلب موارد چنین است)، بنابراین  $\hat{B} = B$ . سپس، ما می توانیم سایش  $A$  را به سادگی با اتساع زمینه آن بدست آوریم (یعنی اتساع  $AC$ ) با همان عنصر ساختاری و تکمیل کننده نتیجه. نظرات مشابه در مورد معادله اعمال می شود.

(۹-۹). ما به طور رسمی اعتبار معادله (۸-۹) را اثبات می کنیم. به منظور نشان دادن رویکردی معمولی برای اثبات اعتبار عبارات شکل شناسی. از تعریف سایش شروع می شود و نتیجه می شود

$$(A \ominus B)^c = \{z | (B)_z \subseteq A\}^c$$

اگر مجموعه  $z$  در  $(B)$  در  $A$  موجود باشد، نتیجه می شود که  $(B)_z \cap A^c = \emptyset$ ، در این صورت موارد قبلی بیان می شود

$$(A \ominus B)^c = \{z | (B)_z \cap A^c = \emptyset\}^c$$

$$(A \ominus B)^c = \{z | (B)_z \cap A^c \neq \emptyset\} \\ = A^c \oplus \hat{B}$$

جایی که آخرین مرحله از تعریف اتساع در معادله (۹-۶) ناشی می شود. و شکل معادل آن در معادله (۹-۷). این اثبات را به پایان می رساند. برای اثبات معادله (۹-۹) می توان از خط استدلال مشابهی استفاده کرد. (به مسئله ۹.۱۶ مراجعه کنید).

### ۹.۳ باز و بسته شدن

همانطور که در قسمت قبل دیدید، اتساع اجزای یک مجموعه را گسترش می دهد و سایش آن را کوچک می کند. در این بخش، ما درباره دو عملیات مهم شکل شناسی دیگر بحث می کنیم: باز و بسته شدن. باز کردن به طور کلی خط یک جسم را صاف می کند، استخوان های باریک را می شکند و برجستگی های نازک را از بین می برد. بستن همچنین باعث صاف شدن بخش هایی از خطوط می شود، اما بر خلاف باز شدن، به طور کلی شکاف های باریک و خلیج های باریک بلند را جوش می دهد، حفره های کوچک را از بین می برد و شکاف های دیگر را پر می کند.

باز شدن مجموعه  $A$  با ساختار عنصر  $B$ ، نشان داده شده با  $A \circ B$ ، به عنوان تعریف شده است

$$A \circ B = (A \ominus B) \oplus B \quad (9-10)$$

بنابراین، دهانه  $A$  توسط  $B$  سایش  $A$  توسط  $B$  است، و پس از آن اتساع نتیجه توسط  $B$  انجام می شود. به طور مشابه، بسته شدن مجموعه  $A$  با ساختار عنصر  $B$ ، نشان دهنده  $A \bullet B$ ، تعریف شده است مانند

$$A \bullet B = (A \oplus B) \ominus B \quad (9-11)$$

که می گوید بسته شدن  $A$  توسط  $B$  فقط اتساع  $A$  توسط  $B$  است و به دنبال آن سایش نتیجه توسط  $B$ . معادله (۹-۱۰) دارای یک تفسیر هندسی ساده است: باز کردن  $A$  توسط  $B$  اتحاد همه تبدیل های  $B$  است به طوری که  $B$  به طور کامل در  $A$ . شکل ۹.۸ (الف) شکل حاوی مجموعه (شی)  $A$  را نشان می دهد. و شکل ۹.۸ (ب) یک سازه جامد و دایره ای است، ب. شکل ۹.۸ (ج) برخی از تبدیل های  $B$  را به گونه ای نشان می دهد که