2
Compressão

2.1
Compressão de Imagens

Na transmissão de imagens, procurando diminuir o uso da largura de banda é usado um método de compressão ou codificação da fonte onde a informação original é mapeada em um novo conjunto, algumas vezes com diferente representação mas de menor tamanho. Este conjunto pode conter a totalidade da informação original, no caso de compressão sem perdas, ou apenas a parte mais significativa. No caso de compressão com perdas é usado o fato da informação redundante poder ser eliminada sem um detrimento apreciável na qualidade visual da imagem recuperada [1].

Existem na atualidade técnicas de compressão de imagens com diversas características que oferecem vantagens dependendo basicamente da aplicação, por exemplo, na transmissão de imagens através de um canal ruidoso é usado geralmente para a compressão técnicas como JPEG(*Joint Picture Expert Group*) que são combinadas com um código convolucional ou um código Reed Solomon para correção de erros com o propósito de melhorar o desempenho do sistema. A Figura 2.1, ilustra esta combinação.

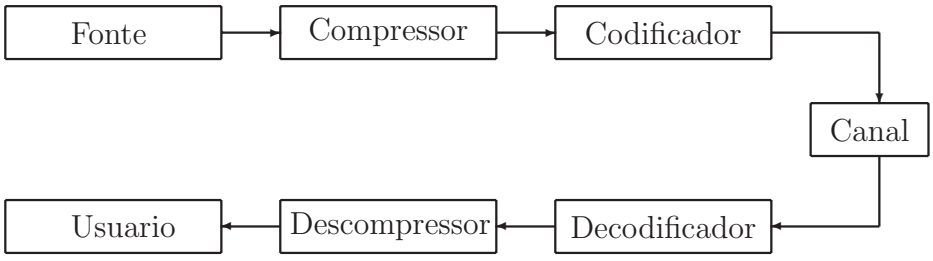

Figura 2.1: Diagrama em Blocos de um Sistema de comunicações para Transmissão de Imagens.

Técnicas baseadas na decomposição em wavelet como EZW (*Embedded Zero Tree Wavelets*) [12] com bom desempenho em qualidade visual para baixas taxas de compressão quando usadas em canais ruidosos, são muito

sensíveis a erros e uma descompressão catastrófica ocorre se existem erros no começo do *bitsream* decodificado, A Figura 2.2 ilustra esta situação, mas aqui o uso de apropriados Códigos FEC (*forward Error Correction*) e trabalhos feitos com soluções para correção de erros que propõem uso de Técnicas UEP(*Unequal Error Protection*) [13] podem ajudar para seu aproveitamento.

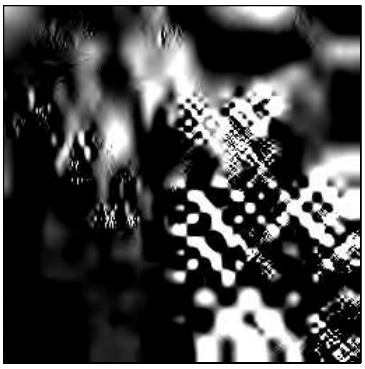

Figura 2.2: Lena 256×256 comprimida com SPIHT logo de que o *bitstream* atravessa um canal AWGN com $E_b/N_o = 6dB$ a imagem é descomprimida, na imagem da direita os erros ocorrem no início do bitstream.

Neste trabalho a técnica usada para a compressão de imagens é a técnica de compressão SPIHT (*Set Partitioning In Herichical Trees*) [10] baseada na técnica EZW. Estas técnicas, de transmissão progressiva, têm a característica de proporcionalidade direta entre o tamanho do *bitstream* gerado pelo compressor e a qualidade da imagem depois do processo de descompressão, neste esquema a taxa de compressão vem dada pela razão entre o tamanho do *bitstream* comprimido e o tamanho da imagem original . No caso de uma transmissão através de um canal ruidoso com codificação para canal além da quantidade de bits recebidos serão o número de bits decodificados com sucesso no receptor os quais determinaram a qualidade visual da imagem recuperada. Sua característica de ser uma codificação *embedded* permite que o processo de decodificação possa ser interrompido e o de recuperação da imagem

iniciado em qualquer momento, por exemplo, uma vez seja atingida uma certa taxa de distorção, estas técnicas também oferecem uma alta qualidade visual para taxas baixas de compressão.

2.2 SPIHT

Desenvolvida por Said e Pearlman, SPIHT é uma técnica de transmissão progressiva baseada em EZW onde se aplica a transformada Wavelet na imagem que se deseja comprimir, os coeficientes obtidos(valores reais) são convertidos a valores inteiros através de um processo de quantização, processo onde se realiza uma perda da informação original o que faz deste um método de compressão com perdas. A seguir apresentamos os conceitos que se encontram envolvidos e que se precisam para o entendimento da técnica SPIHT.

2.2.1 Transmissão progressiva de imagens

Definindo a imagem original como uma matriz $\mathbf{p}$ de pixels com valores, $p_{i,j}$, onde (i,j) com $i = 1, ...n$, e $j = 1, ...m$, se corresponde as coordenadas do pixel, pode se aplicar sobre este conjunto uma transformação Ω definida assim:

$$\mathbf{c} = \Omega(\mathbf{p}) \tag{2-1}$$

Onde Ω é uma transformação unitária hierárquica em sub-bandas. Em uma transmissão progressiva o decodificador depois de receber o valor aproximado ou exato de alguns coeficientes, $\hat{c}$, poderá obter a imagem reconstruída.

$$\hat{\mathbf{p}} = \Omega^{-1}(\hat{\mathbf{c}}) \tag{2-2}$$

O propósito em uma transmissão progressiva é selecionar a informação mais relevante procurando obter o mínimo de distorção. Para medir esta distorção vamos usar o parâmetro mse (*mean square error*), sendo $N = m \times n$ o número de pixels na imagem tem-se:

$$D_{mse}(\mathbf{p} - \hat{\mathbf{p}}) = \frac{\|\mathbf{p} - \hat{\mathbf{p}}\|^2}{N} = \frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{m}(p_{i,j} - \hat{p}_{i,j})^2 \tag{2-3}$$

Usando o fato que a norma euclidiana é invariante em uma transformação unitária neste caso Ω, podemos escrever

$$D_{mse}(\mathbf{p} - \hat{\mathbf{p}}) = D_{mse}(c - \hat{c}) = \frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{m}(c_{i,j} - \hat{c}_{i,j})^2 \tag{2-4}$$

Da equação (2-4) pode se notar que o mse diminui em $\frac{|c_{i,j}-\hat{c}_{i,j}|^2}{N}$ assim coeficientes com maior magnitude devem ser transmitidos primeiro por que eles tem maior quantidade de informação no sentido a que eles conseguem uma diminuição maior da distorção. Além disto a informação contida no valor $|c_{i,j}|$ pode ser ordenada de acordo a sua representação binária e os bits mais significativos transmitir-se primeiro. O processo de transmissão progressiva se fundamenta nestes dois conceitos básicos, ordenação de coeficiente por magnitude e prioridade na transmissão dos bits mais significantes.

2.2.2 Transmitindo os valores dos coeficientes

Assumamos que os coeficientes são ordenados de acordo com o número mínimo de bits requeridos para sua representação binária, isto é fazendo um mapeamento um por um $\eta : I \mapsto I^2$ tal que: $\lfloor \log_2 |c_{\eta(k)}| \rfloor \geq \lfloor \log_2 |c_{\eta(k+1)}| \rfloor$

A Figura 2.3 mostra um esquema de representação binária de uma lista de coeficientes em ordem de magnitude, cada coluna k contem os bits de $c_{\eta(k)}$. Os bits na fila superior indicam o sinal do coeficiente, as filas serão numeradas tal que a parte superior corresponde aos bits mais significativos (msb) e os bits da fila inferior serão então os bits menos significativos (lsb).

Agora assumamos que além do ordenamento da informação o decodificador recebe o número μ_n de coeficientes tais que $2^n \leq |c_{i,j}| < 2^{n+1}$. Na Figura 2.3, por exemplo, se tem $\mu_5 = 2$, $\mu_4 = 2$, $\mu_3 = 4$, etc. Como Ω é uma transformação unitária todos os bits em uma fila tem o mesmo conteúdo de informação então o método mais efetivo para enviar a informação progressivamente em ordem de magnitude será enviando cada fila seqüencialmente como indicam as setas na Figura 2.3 . Note-se que já que os coeficientes estão ordenados em ordem de magnitude os primeiros "0" e o primeiro "1" de quaisquer coluna não precisa ser transmitido já que pode ser inferido por μ_n e pelo ordenamento. O método de transmissão progressiva pode implementar-se com o uso do seguinte algoritmo:

1. enviar $\lfloor \log_2(max_{(i,j)} \ |c_{(i,j)}|) \rfloor$ ao decodificador
2. enviar μ_n seguido das coordenadas do pixel $\eta(k)$ e do sinal de cada μ_n tal que $2^n \leq |c_{\eta(k)}| < 2^{n+1}$ este passo é denominado Classificação
3. enviar o n-ésimo bit mais significante de todos os coeficientes com $|c_{\eta(k)}| < 2^{n+1}$. Ou seja, aqueles que tiveram coordenadas transmitidas em classificações anteriores, este passo é denominado refinamento

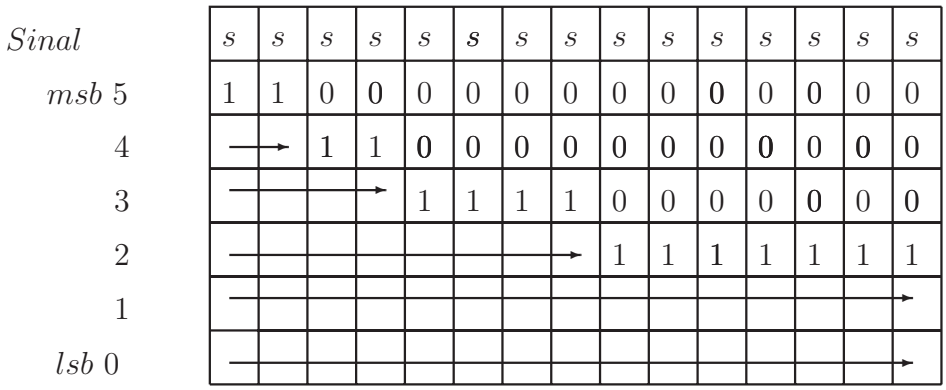

Figura 2.3: Representação binária de coeficientes em ordem de magnitude .

4. fazer $n = n - 1$ e retornar ao passo 2.

O algoritmo para quando é atingida a taxa de distorção desejada.

2.2.3 Classificando os conjuntos particionados

Uma das principais características deste método é que os dados ordenados não são explicitamente transmitidos. Sabe-se que o caminho de execução de um algoritmo se baseia nos resultados de comparações em seus pontos de ramificação. Assim, se o codificador e o decodificador têm o mesmo procedimento de classificação o decodificador pode repetir o caminho de execução do codificador necessitando apenas os dados resultados das comparações realizadas pelo codificador.

Um fato importante usado pelo algoritmo é que não é necessário classificar todos os coeficientes. Basta simplesmente selecionar os coeficientes para os quais $2^n \leq |c_{\eta(k)}| < 2^{n+1}$, com n diminuindo a cada passo. Diz-se que se $|c_{\eta(k)}| < 2^{n+1}$, para um dado n, o coeficiente é insignificante, no caso contrário, que é significante

Aqui o algoritmo de classificação divide o conjunto de coeficientes em subconjuntos particionados γ_m e avalia verificando a desigualdade:

$$\max_{(i,j)\in\gamma_m} |c_{i,j}| \geq 2^n? \tag{2-5}$$

Se a desigualdade é falsa , então o decodificador sabe que todos os coeficientes em γ_m, são insignificantes. Se pelo contrário, a resposta é sim, certa regra compartilhada pelo codificador e decodificador será usada para particionar γ_m em l novos subconjuntos $\gamma_{m,l}$ e o teste de significância será aplicada a estes novos subconjuntos.

Para diminuir o número de comparações define-se uma regra que usa um ordem esperado na hierarquia definida pela pirâmide das sub-bandas. O objetivo é criar uma nova partição onde os conjuntos insignificantes contenham um grande número de elementos e os conjuntos não significantes tenham um único elemento. Para indicar a significância de um conjunto de coeficientes γ se faz-se uso da função indicadora de significância de $c_{i,j}$,

$$\mathcal{S}_n(\gamma) = \begin{cases} 1, & \text{para } \max\limits_{(i,j)\in\gamma_m} |c_{i,j}| \geq 2^n \\ 0, & \text{outro caso} \end{cases} \tag{2-6}$$

2.2.4 Orientação espacial da árvore

Normalmente a energia da imagem transformada fica concentrada nas componentes de baixa freqüência, por este fato quando nos movimentamos ao longo dos conjuntos de coeficientes de sub-bandas a partir das sub-bandas de freqüências mais altas, para as mais baixas dos níveis da pirâmide de sub-bandas a variância diminui. Na Figura 2.4, pode-se observar a orientação espacial da árvore definido-se para uma pirâmide com quatro sub-bandas. cada nó da árvore, correspondente a um coeficiente, é identificado pelas coordenadas do pixel. Isto é descendentes diretos correspondem a pixels com a mesma orientação espacial no mais próximo nível da pirâmide. A arvore é definida tal que cada nó diferente dos descendentes diretos tem a sua vez quatro descendentes (folhas), cada descendente é formado por um grupo de 2×2 pixels adjacentes. Na Figura 2.4, as setas estão orientadas dos nós pais para seus 4 descendentes. Os coeficientes na parte superior da pirâmide formam o nó raiz e formam também um grupo de 2×2 pixels adjacentes. Embora sua regra de ramificação não seja a mesma, em cada grupo deles um não tem descendência, a seguir se mostram os conjuntos de coordenadas que serão usados para explicar o método de codificação:

- $\mathcal{O}(i,j)$: Conjunto de descendentes diretos do nó (i,j)

- $\mathcal{D}(i,j)$: Conjunto de coordenadas de todos os descendentes do nó (i,j)
- $\mathcal{H}$: Conjunto de coordenadas correspondente aos nó raiz da arvore (nós que se encontram no mais alto nível da pirâmide)
- $\mathcal{L}(i,j) = \mathcal{D}(i,j) - \mathcal{O}(i,j)$

por exemplo, exceto para os níveis mais alto e mais baixo da pirâmide tem-se:
$\mathcal{O}(i,j) = \{(2i,2j),(2i,2j+1),(2i+1,2j)(2i+1,2j+1)\}$

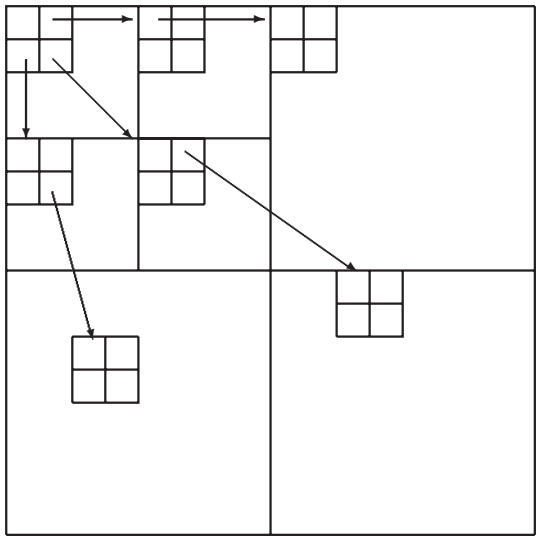

Figura 2.4: Orientação espacial da árvore.

Usa-se parte da orientação espacial para definir a partição de subconjuntos o algoritmo é o seguinte:

Algoritmo para partição dos conjuntos

1. A partição inicial é formada pelo conjunto$\{(i,j)\}$ e $\mathcal{D}(i,j) \forall\ (i,j) \in \mathcal{H}$
2. Se $\mathcal{D}$(i,j) é significante então ele é particionado em quatro conjuntos $\mathcal{L}$(k,l), onde $(k,l) \in \mathcal{O}(i,j)$
3. Se $\mathcal{L}$(i,j) é significante então ele é particionado em quatro conjuntos $\mathcal{D}$(k,l), onde $(k,l) \in \mathcal{O}(i,j)$

2.3 Algoritmo de codificação SPIHT

O primeiro passo para a codificação consiste em criar um mapa de significância para cada limiar establecido. Este mapa possui informação se um elemento da partição se encontra acima do limiar ou não. O mapa de significância se obtém usando a árvore de orientação espacial (relação de descendencia entre os coeficientes wavelet).

O seguinte passo consiste na transmissão dos bits mais significativos mediante duas operações: ordenação e refinamento. Para a implementação do algoritmo usam-se três listas : lista de pixels não significativos (LIP), lista de coeficientes significativos (LSP) e lista de coordenadas não significativas (LIS) . Ao final de cada passo da ordenação, LSP contem as coordenadas de todos os coeficientes significativos para o η correspondente. Também inclui os coeficientes encontrados nos passos anteriores. Os elementos de LIS são coordenadas de coeficientes rotuladas como coordenadas de tipo $\mathcal{A}$ ou $\mathcal{B}$. O rotulo é de tipo $\mathcal{A}$ quando representa todos os descendentes e é do tipo $\mathcal{B}$ quando representa a todos os descendentes a partir dos netos.

O seguinte é o algoritmo usado pelo SPIHT:
Algoritmo de codificação SPIHT

1. **Inicialização** enviar: $n = \lfloor \log_2(|max_{i,j}\{|c_{i,j}|\})\rfloor$. A LSP está vazia e se levam todas as coordenadas $(i,j) \in \mathcal{H}$ a LIP e ao conjunto resultante une-se o conjunto de todos os seus descendentes se adicionam a LIS como elementos do tipo $\mathcal{A}$.
2. **Classificação**:
   - 2.1 Para cada entrada (i,j) na LIP fazer:
     - 2.1.1 Obter $\mathcal{S}_n(i,j)$
     - 2.1.2 Se $\mathcal{S}_n(i,j)$=1 então: mover (i,j) para a LSP e obter o sinal de $c_{i,j}$;
   - 2.2 para cada entrada (i,j) na LIS fazer:
     - 2.2.1 se a entrada é do tipo $\mathcal{A}$ então:
       - ⋆ saída $\mathcal{S}_n(\mathcal{D}(i,j))$

⋆ se $\mathcal{S}_n(\mathcal{D}(i,j)) = 1$ então:

· Para cada $(k,l) \in \mathcal{O}(i,j)$:
Obter $\mathcal{S}_n(k,l)$
- se $\mathcal{S}_n(k,l)$=1 então, agregar (k,l) à LSP e calcular o sinal de $c_{k,l}$;
- se $\mathcal{S}_n(k,l)$=0 então, agregar (k,l) ao final de LIP;

· se $\mathcal{L}(i,j) \neq \emptyset$, então mover$(i,j)$ para o final de LIS como entrada tipo $\mathcal{B}$, e avançar para o passo 2.2.2, caso contrário remover a entrada(i,j) de LIS.

- 2.2.2 se a entrada é de tipo $\mathcal{B}$ então: mover (i,j) à LSP e sai o sinal de $c_{i,j}$

- obter $\mathcal{S}_n(\mathcal{L}(i,j))$.

- Se $\mathcal{S}_n(\mathcal{L}(i,j)) = 1$ então.

⋆ agregar cada $(k,l) \in \mathcal{O}(i,j)$ao final de LIS como entrada de tipo $\mathcal{A}$.

⋆ remover (i,j) de LIS.

3. **Refinamento**: Para cada entrada (i,j) de LSP, exceto aqueles incluídos nas etapas anteriores (ou seja com o mesmo n), calcular o n-ésimo msb de $|c_{i,j}|$;

4. fazer $n = n - 1$ e retornar ao passo 2.

A seguir é mostrado um exemplo donde se seguem os três mais importantes passos que usa o algoritmo SPIHT.

2.4 Exemplo

2.4.1 Inicialização

No passo de inicialização, n se obtém de calcular $\log_2 C_{max}$, onde C_{max} é o coeficiente maior em valor absoluto, obtido da matriz de coeficientes, o limiar em cada passo da classificação será 2^n com n diminuindo em 1. Inicialmente a LSP esta vazia, à LIP agregam-se as coordenadas dos coeficientes do nível

mais alto e LIS as coordenadas dos coeficientes raiz como tipo $\mathcal{A}$, ver Figura 2.5.

2.4.2 Ordenação

A ordenação consiste em verificar se cada elemento de tipo $\mathcal{A}$ em LIP é ou não significante para o valor corrente de n. Se o elemento é significante transmite se "1", seguido do bit "0"ou "1"para indicar o sinal do coeficiente, para logo levar suas coordenadas a LSP. Se não é significante transmite-se o bit "0". Em seguida verifica-se a significância dos descendentes de cada entrada de LIS. Se não há elemento significante transmite se um "0", no caso contrário "1", e de novo se verifica a significância de cada membro de sua descendência . Sim se trata de elemento significante agrega-se a LSP e transmite-se seu sinal e, caso contrário adicione se a LIP e transmite se um bit "0". Se esse coeficiente dispõe de mais descendentes, colocam se suas coordenadas ao final de LIS e se marca como tipo $\mathcal{B}$. Por o contrário, se o elemento LIS é de tipo $\mathcal{B}$, verifica se tem descendentes significativos a partir dos netos (incluídos). Sim se confirma se transmite "1"e se adiciona seus coordenadas correspondentes ao final de LIS marcadas como tipo $\mathcal{A}$. No caso contrário se transmite "0"e se eliminam suas coordenadas de LIS. As entradas adicionadas a LIS não se tem em conta na etapa posterior ao refinamento.

2.4.3 Refinamento

O refinamento consiste em avaliar os componentes da LSP introduzidos nas etapas anteriores, enviando o n-ésimo bit mais significativo . Por último se diminui o limiar em "1"e se volta ao passo da ordenação. O ciclo se repete até alcançar o limiar zero, incluindo ele. O resultado do algoritmo consiste em um vetor composto por zeros e uns, que serão armazenados em um arquivo, a este vetor nos referimos como *bitstream*. O número de elementos neste mapa determina o fator de compressão proporcionado pelo algoritmo para a imagem dada. A seguir são mostrados os esquemas nas diferentes etapas de execução do algoritmo.

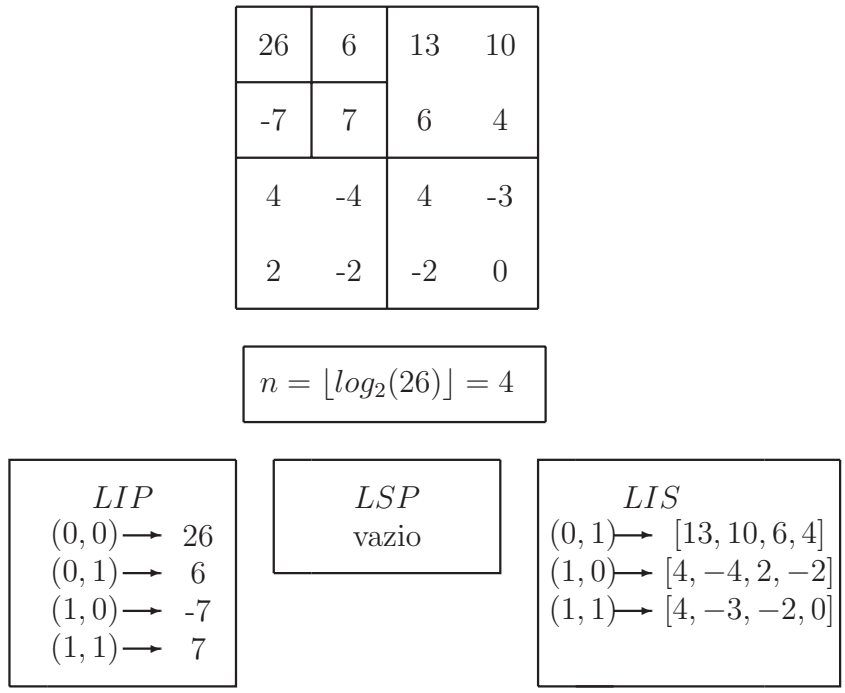

Figura 2.5: Esquema de Inicialização

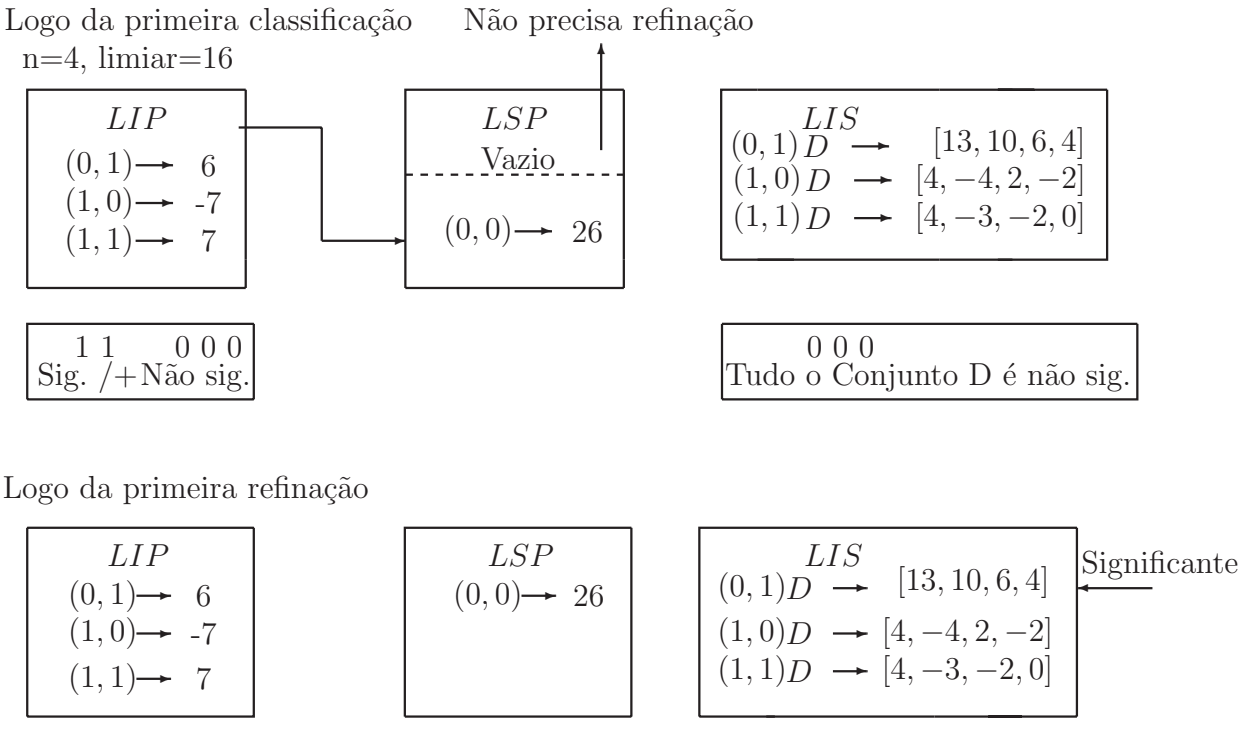

Figura 2.6: Esquema da Primeira classificação

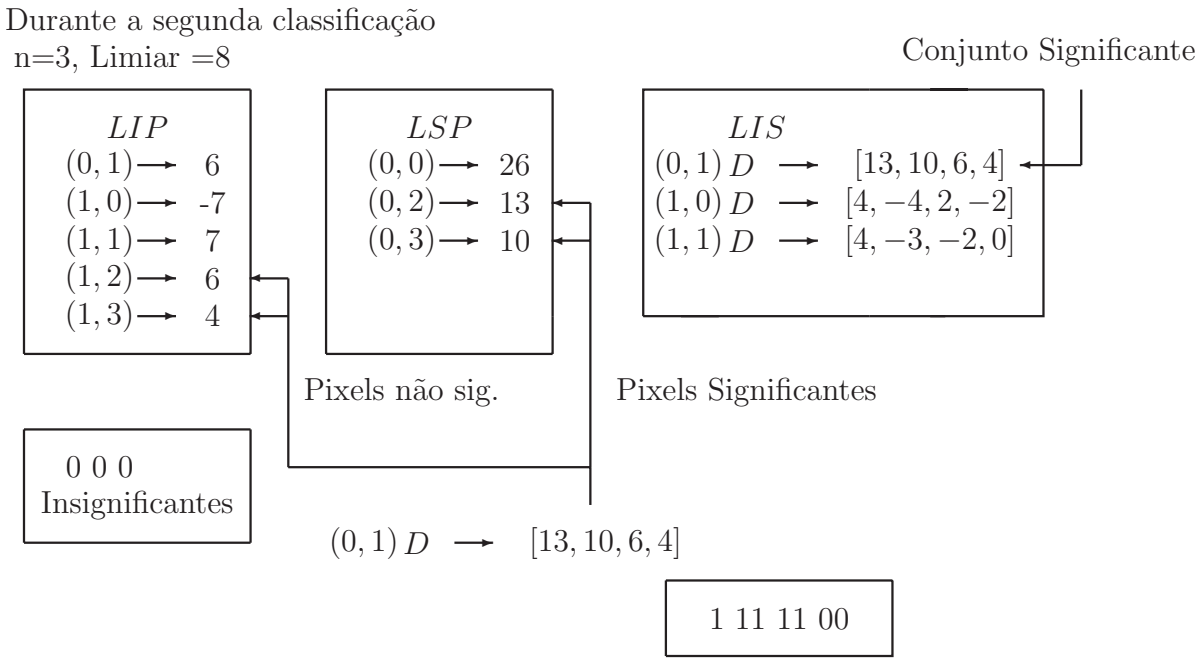

Figura 2.7: Esquema de segunda classificação

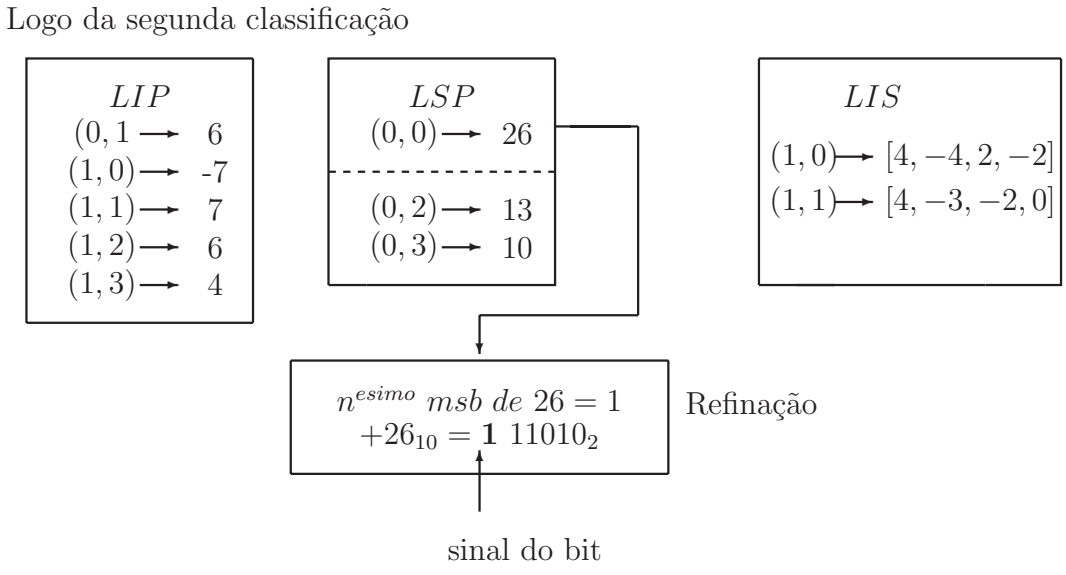

Figura 2.8: Esquema do refinamento

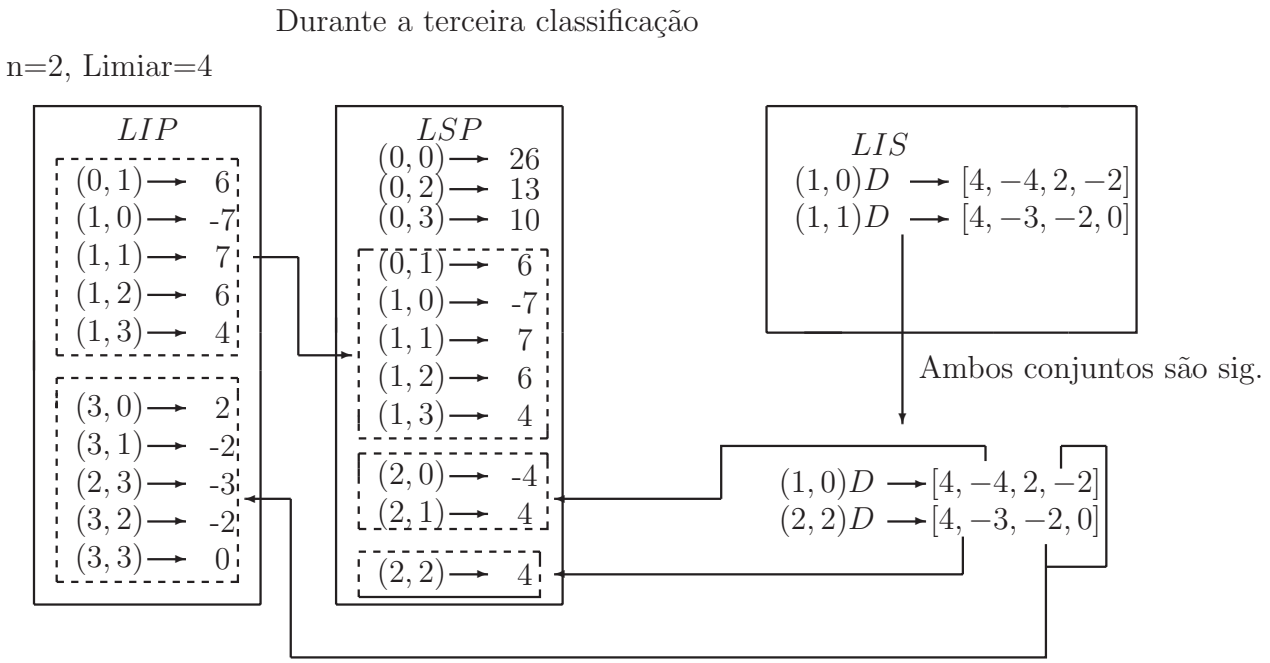

Figura 2.9: Esquema da terceira classificação

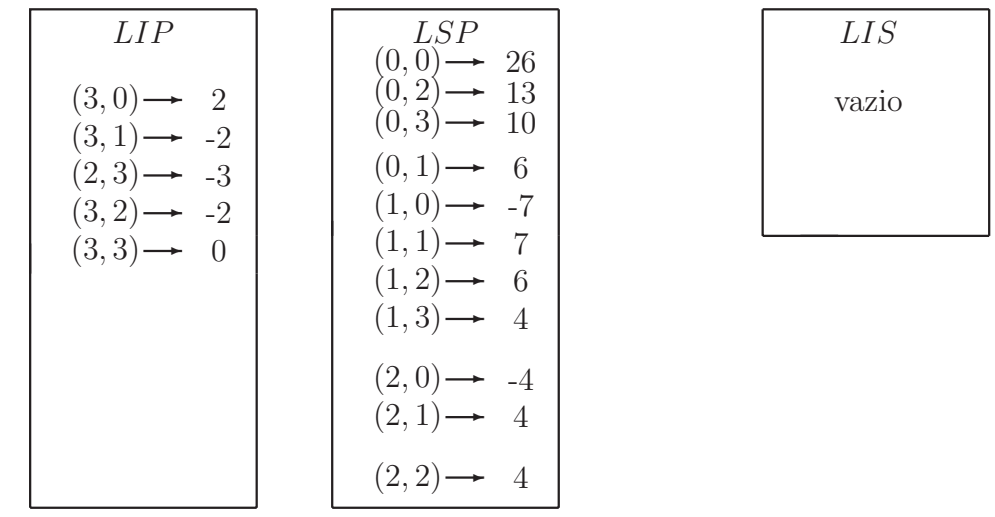

Figura 2.10: Esquema logo da terceira classificação

A avaliação de desempenho de um esquema de compressão qualquer pode ser feita objetivamente, através da medida PSNR (*Peak Signal to Noise Ratio*), ou subjetivamente através da inspeção visual das imagens. A medida PSNR é útil por que nos fornece uma base matemática de análise. Ela se baseia em medir a diferença entre o valor do pixel na imagem original, x_i, e seu valor na reprodução da imagem após da descompressão, $\hat{x}_i$. Os $\mathcal{N}$ pixels da imagem são considerados. A expressão para o cálculo do valor PSNR em dB pode ser vista na equação

$$PSNR = 10log_{10}\left[\frac{255^2}{\frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}}(x_i - \hat{x}_i)}\right] \tag{2-7}$$

A Figura 2.11, mostra o desempenho deste algoritmo medido através do parâmetro PSNR em imagens usadas ao longo deste trabalho, outras características importantes que possui esta técnica são sua rapidez e possível uso em receptores com diferentes taxas, ou seja, onde se possa trabalhar com uma qualidade da imagem diferenciada.

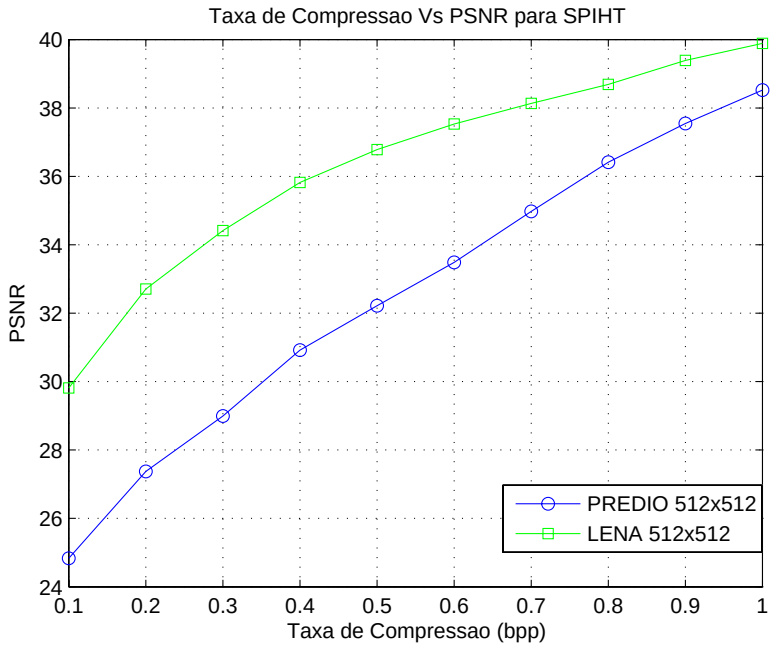

Figura 2.11: Desempenho do Algoritmo SPIHT.

As técnicas de compressão baseadas em decomposição em wavelet representaram um dos avanços mais importantes no processamento digital de imagens, características tais como qualidade, rapidez , simplicidade entre outras tem feito que estas técnicas de compressão venham sendo usadas ampla-

mente em pesquisa, seu uso em versões modernas para compressão de imagens tais como $JPEG2000$ mostram a atualidade e eficácia que esta técnica apresenta.

2.5 Resumo

Neste capítulo foi mostrada a importância que têm o processo de compressão para um bom uso dos recursos em um sistema de transmissão de imagens, foi apresentado a técnica SPIHT a qual foi escolhida para o desenvolvimento deste trabalho, foram descritas as características mais importantes, suas vantagens e também os problemas que apresenta quando erros ocorrem no começo do *bitstream* gerado na compressão, foi descrito o algoritmo que esta técnica usa, mostrou-se um exemplo que ensina cada um dos passos que segue o algoritmo e através de simulação mostrou-se o desempenho medido com o parâmetro PSNR em duas imagens, Prédio e Lena.