

dos $n \times n$ não sobrepostos; cada bloco é processado individualmente. Esses blocos são passados então pela transformada direta para gerar $(N/n)^2$ sub-imagens $n \times n$ transformadas. Como isto, descorrelaciona-se os pixels de cada sub-imagem, compactando-se o máximo possível a informação em um número menor de coeficientes de transformação. A quantização encarrega-se de eliminar ou quantizar seletivamente os coeficientes que carregam menos informações. A codificação de símbolos gera a imagem comprimida aplicando-se um código de tamanho variável.

O decodificador realiza o processo inverso e consiste em quatro componentes básicos: decodificação de símbolos, transformação inversa, desquantização e união das sub-imagens. A Figura 2.7(b) mostra esses componentes básicos.

O padrão mais popular de codificação de imagens por transformada é o Joint Photographers Expert Group (JPEG), que utiliza a transformada cosseno discreta [22, 23].

Transformada cosseno discreta (DCT)

A transformada cosseno discreta é excelente em termos de compactação de energia.

A transformada cosseno discreta $C(u)$ e a inversa da transformada cosseno discreta $f(x)$ unidimensional são obtidas pelas Equações (2.14) e (2.15), respectivamente [15].

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \quad (2.14)$$

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \quad (2.15)$$

onde a função α é:

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & \text{para } u=0 \\ \sqrt{\frac{2}{N}} & \text{para } u=1, 2, 3, \dots, N-1 \end{cases}$$

A transformada cosseno discreta bidimensional $C(u, v)$ e sua inversa são obtidas pelas Equações (2.16) e (2.17), respectivamente [15]:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (2.16)$$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(x)\alpha(y) C(u, v) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (2.17)$$

onde as funções $\alpha(u)$ e $\alpha(v)$ são:

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & \text{para } u=0 \\ \sqrt{\frac{2}{N}} & \text{para } u=1, 2, 3, \dots, N-1 \end{cases}$$

$$\alpha(v) = \begin{cases} \frac{1}{\sqrt{N}} & \text{para } v=0 \\ \sqrt{\frac{2}{N}} & \text{para } v=1, 2, 3, \dots, N-1 \end{cases}$$

2.5.4 Codificação em sub-bandas

A codificação em sub-bandas baseia-se no princípio da decomposição do sinal de entrada (imagem) em múltiplas sub-bandas através de um conjunto de filtros passa-faixa, onde cada sub-banda sofre o processo de decimação responsável pela definição da sua largura. Cada sub-banda é codificada separadamente.

A vantagem da utilização do esquema de codificação em sub-bandas é que com uma apropriada alocação de bits nas diferentes sub-bandas, de acordo com um critério percentual estabelecido (decimador), o número de níveis de quantização e a variação do erro de reconstrução podem ser controlados separadamente em cada sub-banda.

Na decodificação, as sub-bandas codificadas são decodificadas, interpoladas, filtradas pelo banco de filtros correspondentes e, finalmente, recompostas para reconstruir a imagem original.

Um exemplo de compressão por sub-bandas que tem despertado grande

interesse pela sua utilidade e pelo seu potencial é a wavelet. Em 1985, Stephane Mallat descobriu algumas relações entre os filtros, algoritmos piramidais e bases wavelet continuamente diferenciáveis. Dois anos mais tarde, Ingrid Daubechies usou os trabalhos de Mallat para construir um conjunto de funções base ortonormais wavelet, que estabeleceu a base sólida para as atuais aplicações wavelets [24].

2.5.5 Codificação por quantização vetorial

A codificação por quantização vetorial baseia-se na divisão da imagem em blocos quadrados ou retangulares de pixels de tamanho fixo, chamados vetor imagem ou vetor alvo. Cada vetor imagem é codificado como um vetor código, que está associado a uma lista de vetores código, chamada de dicionário ou codebook. O dicionário contém um conjunto de vetores padrões de tamanho fixo, disponível para a codificação e para a decodificação. O armazenamento da imagem consiste em uma lista de códigos que indica qual vetor código no dicionário melhor combina com a sequência de vetores alvo que compõem a imagem. A decodificação subsequente consiste simplesmente na colagem ou na montagem dos vetores códigos indicados pelos índices na lista dos vetores código [12, 25].

2.5.6 Codificação por fractais

As técnicas de compressão com perdas estudadas até aqui neste capítulo realizam a compressão eliminando ou reduzindo alguma redundância de dados da imagem, não resultando em altas taxas de compressão. A técnica baseada em fractais explora as auto-similaridades existentes nas imagens e

proporciona uma taxa de compressão bem mais alta do que as obtidas por outras técnicas.

A técnica de compressão por fractais foi criada e patenteada por Michael F. Barnsley por volta de 1987. Em 1989, Arnald Eric Jacquin publicou sua tese [26], sendo o primeiro trabalho de divulgação do esquema de compressão de imagens por fractal [19]. Este trabalho despertou o interesse de diversos pesquisadores. Na década de 90, dezenas de pesquisas em compressão por fractal foram desenvolvidas e publicadas, visando, principalmente, diminuir o tempo do processo de codificação e melhorar a fidelidade da imagem reconstruída. Apesar dos avanços significativos obtidos, estas duas questões ainda são objetos de pesquisas.

A compressão de imagem por fractal é baseada nos conceitos e resultados dos sistemas de funções iteradas particionadas (partitioned iterated function systems - FIPS), que parte regiões de uma imagem e as mapeiam em regiões menores. Estes sistemas buscam a auto-similaridade entre as partes maiores e as partes menores da imagem. Desta forma, as imagens são consideradas como uma colagem de partes auto-similares que podem ser mapeadas entre si. Como partes menores considera-se blocos quadrados, de dimensões $n \times n$ pixels, chamados de blocos de imagem ou molde, e como partes maiores considera-se blocos quadrados, com o dobro das dimensões ($2n \times 2n$ pixels), chamados de blocos domínio.

A compressão e descompressão por fractal são processos inerentemente assimétricos, pois, o processo de codificação da imagem é de computação intensiva e lento, enquanto o processo de decodificação é bem mais simples e de execução rápida. Essa assimetria é uma vantagem real da com-

pressão fractal sobre os métodos simétricos JPEG e outros, porque, na maioria das aplicações, a compressão da imagem ocorre uma vez, no momento do armazenamento da imagem codificada, enquanto a descompressão ocorre inúmeras vezes quando a imagem é exibida.

Os contínuos avanços e aperfeiçoamentos das técnicas de compressão por fractal têm permitido obter altas taxas de compressão, com perdas visualmente imperceptíveis na qualidade da imagem reconstruída.

2.6 Método de Compressão Joint Photographic Experts Group (JPEG)

Um comitê conjunto formado pelo Consultative Committee of International Telephone and Telegraph (CCITT) e pelo International Standardization Organization (ISO) desenvolveram o método de compressão Joint Photographic Experts Group (JPEG). Este método acabou tornando-se um padrão de referência. As vantagens desse padrão fez com que o seu uso crescesse rapidamente na Internet e no armazenamento de imagens.

Existem quatro modos de operação do padrão JPEG: seqüencial, progressivo, hierárquico e sem perdas. [2]

Como as imagens utilizadas nesta dissertação são imagens com níveis de cinza e o modo de operação utilizado nelas é o seqüencial, é explicado somente esse modo neste capítulo.

Os principais passos do padrão JPEG são: divisão da imagem em blocos, deslocamento dos níveis de cinza, aplicação da DCT, quantização e codificação.

Divisão da imagem em blocos

No JPEG, a imagem a ser comprimida é dividida em blocos não sobrepostos de tamanho 8 x 8 pixels.

Deslocamento dos níveis de cinza

Após a divisão da imagem em blocos. Em cada bloco, os valores dos pixels são subtraídos de 2^{n-1} , onde 2^n é o número de níveis de cinza da imagem. Isso altera os valores positivos (de 0 a $2^n - 1$) para números inteiros na faixa de -2^{n-1} a $2^{n-1} - 1$. Por exemplo, se o número de níveis de cinza da imagem é igual a 256, o valor de cada nível de cinza é subtraído de $2^7 = 128$, isso altera os valores positivos (de 0 a 255) para números inteiros na faixa de -128 a 127.

Aplicação da DCT

Após alterar os níveis de cinza de cada bloco, calcula-se a DCT de cada um deles. Dos 64 coeficientes de cada bloco muitos possuem valores muito pequenos próximos de zero. Assim, existe menos de 64 coeficientes significativos. Quanto menos desses coeficientes existirem, maior será a compactação.

No processo de quantização ainda ocorrerá diminuição no número de coeficientes significativos da DCT.

Quantização

Na quantização, a matriz de cada bloco é dividida pela matriz de quantização de luminância JPEG, mostrada na Tabela 2.1. E os valores

resultantes são arredondados para números inteiros. Nessa divisão o valor de vários coeficientes significativos é muito pequeno e próximo de zero.

Tabela 2.1: Matriz de quantização de luminância típica do padrão JPEG.

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
19	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Ordenação zigue-zague

Após arredondar todos os coeficientes de cada bloco da imagem, para que os coeficientes mais significativos fiquem apenas na parte superior de cada bloco, é feita uma ordenação conforme o padrão zigue-zague, mostrado na Figura 2.8. Essa ordenação é feita para aproveitar as longas seqüências de zero no processo de quantização.

Após a ordenação zigue-zague, os maiores coeficientes estão localizados na parte superior e os menores coeficientes estão localizados na parte inferior da matriz. A maior parte desses últimos coeficientes possuem valores muito próximos ou igual a zero.

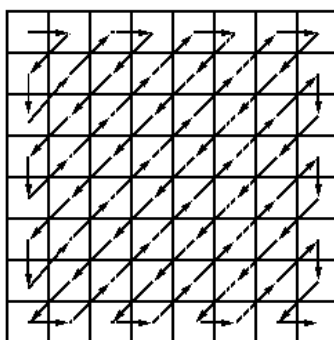


Figura 2.8: Padrão de ordenação zigue-zague utilizado no método de compressão JPEG.

Codificação

Após a quantização de cada bloco, começa a codificação de cada um deles. O primeiro coeficiente (nível dc) de cada bloco é codificado utilizando a codificação DPCM, e o restante dos coeficientes são codificados pela codificação RLE como explicado na sub-seção 2.4.1 deste capítulo. É na codificação que os blocos são seriados, ou seja, ao invés deles serem tratados como matrizes, eles passam a ser tratados como seqüências de números, na transmissão ou no armazenamento.

Na DPCM utilizada no padrão JPEG, o primeiro coeficiente é transmitido e ele é usado como valor previsto para todos os blocos da imagem. Daí em diante não são transmitidos os primeiros coeficientes dos outros blocos e sim a diferença entre os valores reais do primeiro coeficiente do primeiro bloco e do primeiro coeficiente do bloco em questão.

O último passo do processo de compressão do padrão JPEG é a codificação de Huffman descrita na sub-seção 2.4.2 deste capítulo.

A descompressão do padrão JPEG é o processo inverso da compressão.

A codificação das matrizes de quantização obtidas neste trabalho também foram realizadas da mesma maneira explicada no padrão JPEG.

A quantização escalar é um passo importante no padrão JPEG e na obtenção da matriz de quantização a ser encontrada neste trabalho, por isso ela é descrita a seguir.

Quantização escalar

O quantizador possui um codificador e o desquantizador possui um decodificador, como mostrado na Figura 2.9.

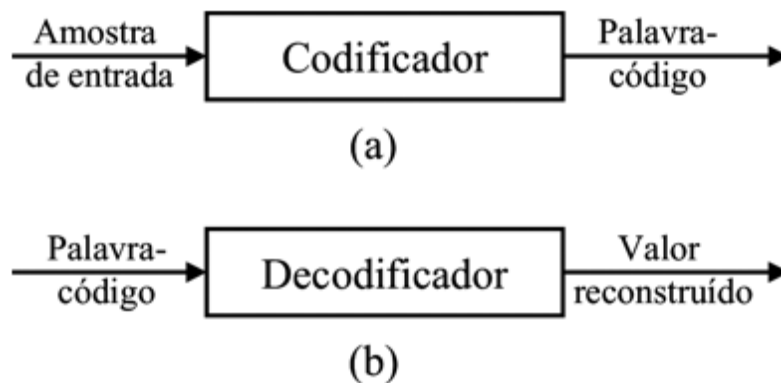


Figura 2.9: (a) quantizador; (b) desquantizador

O codificador divide o conjunto de valores de amostras de entrada em M intervalos e atribui a cada intervalo uma palavra-código (codeword). Os limites entre esses intervalos são denominados de limites de decisão (d_i).

O decodificador atribui um valor de reconstrução para cada palavra-código gerada no codificador. Geralmente esse valor é a média da faixa dos

valores do intervalo. Na prática, o valor reconstruído é obtido em uma tabela pela busca do índice transmitido pelo codificador. O conjunto de todos os possíveis valores de saída dessa tabela é chamado livro-código (codebook). A Tabela 2.2 mostra um exemplo de um livro-código com quatro palavras-código.

Tabela 2.2: Exemplo de um livro-código com 4 palavras-código

Intervalos	Palavra-código
$d_0 < x \leq d_1$	y_0
$d_2 < x \leq d_3$	y_1
$d_4 < x \leq d_5$	y_2
$d_6 < x \leq d_7$	y_3

O operador $Q(\cdot)$ de um quantizador escalar com M intervalos é definido pela Equação (2.18).

$$Q(x) = y_i \quad d_{i-1} < x \leq d_i \quad (2.18)$$

onde:

d_i - limite de decisão, $1 \leq i \leq M$.

y_i - valor reconstruído no codificador, $1 \leq i \leq M$.

O livro-código para um quantizador escalar com M intervalos é mostrado na Equação (2.19).

$$C = \{y_1, y_2, \dots, y_M\} \quad (2.19)$$

O número de bits necessário de um valor de saída do livro-código é obtido pela Equação (2.20).

$$R = \log_2(M) \quad (2.20)$$

O método mais utilizado para medir o erro que ocorre no processo de quantização é o erro quadrático de distorção, obtido pela Equação (2.21).

$$d(x, y_i) = |x - y_i|^2 \quad (2.21)$$

onde:

x - amostra de entrada do codificador.

A medida mais utilizada para medir o desempenho de um quantizador é a média estatística da distorção obtida pela Equação (2.22).

$$D = \int_{-\infty}^{\infty} d(x, \bar{x}) f_x(x) dx \quad (2.22)$$

onde:

$f_x(x)$ – função densidade de probabilidade (pdf) de x .

A média estatística do erro quadrático é o erro quadrático médio (mean square error - MSE), obtido pela Equação (2.23). Esta é outra maneira de escrever a Equação (2.4) deste capítulo.

$$D = \sum_{i=1}^N (x - y_i) f_x(x) dx \quad (2.23)$$

Quanto menor o valor de D , mais o quantizador aproxima-se do quantizador ótimo.

Existem dois tipos de quantizador: uniforme e não-uniforme.

O quantizador uniforme possui intervalos de comprimento constante, como é mostrado na Figura 2.10. Este quantizador é utilizado quando existe a necessidade de uma uniformidade nos resultados, assim para as todas amostras o erro é o mesmo.

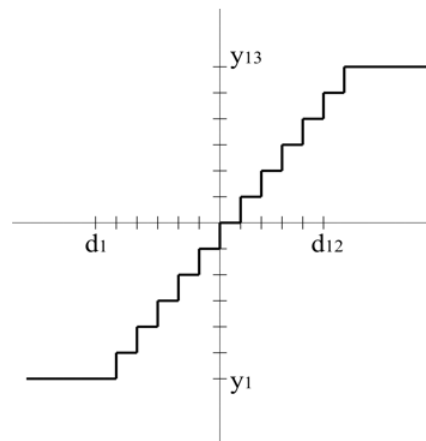


Figura 2.10: Esquema de um quantizador uniforme.

Pode ser observado na Figura 2.10, que o intervalo entre os valores de decisão d_i são constantes.

O quantizador não-uniforme tem intervalos de comprimento variável, isto possibilita quantizar sinais com amplitudes de tamanhos diferentes, ajustando cada valor amostra com um erro diferente. No quantizador não-uniforme os valores de entrada mais significativos são quantizados com maior

precisão e os menos significativos são quantizados com menor precisão. A Figura 2.11 mostra um quantizador não-uniforme.

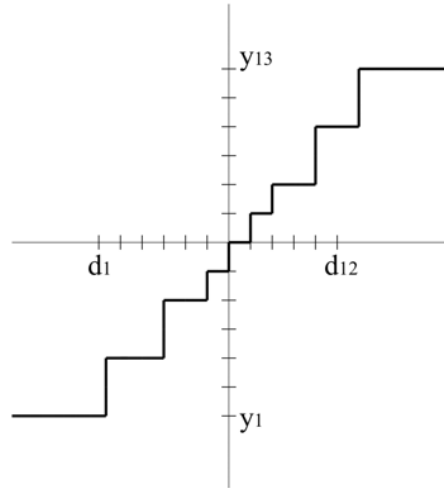


Figura 2.11: Esquema de um quantizador não-uniforme.

Pode ser observado na Figura 2.11, que o intervalo entre os valores de decisão d_i não são constantes.

Neste trabalho utilizou-se o quantizador não-uniforme nos 64 coeficientes da matriz de quantização.

2.7 Considerações Finais deste Capítulo

Este capítulo descreveu as medidas de fidelidade em compressão e as técnicas de compressão sem perdas e com perdas. E finalmente apresentou o padrão de compressão JPEG.

Capítulo 3

MÉTODO UTILIZADO

3.1 Introdução

A quantização é a base da compressão com perdas. A otimização da matriz de quantização é o principal objetivo para se conseguir uma baixa taxa de bits/pixel para uma imagem.

Os parâmetros mais importantes para se avaliar a tendência de um conjunto de valores de níveis de cinza em uma imagem são: a média aritmética, a média geométrica e a moda. Cada situação deve ser analisada para se julgar qual o melhor desses parâmetros a ser utilizado.

Como o intervalo de níveis de cinza em uma imagem é uma progressão aritmética, neste trabalho é utilizado o método estatístico da média aritmética para obter uma matriz de quantização otimizada para imagens médicas gravadas em níveis de cinza.

Este capítulo apresenta o método utilizado neste trabalho para obter a matriz de quantização utilizando a média aritmética e mostra a matriz de quantização obtida por Hamamoto para imagens de ultra-sonografia [1]. Finalmente são realizadas as considerações finais deste capítulo.

3.2 Imagens Utilizadas

O padrão usado na prática para armazenar imagens médicas é o Digital Imaging and Communications in Medicine (DICOM) [27].

As imagens utilizadas neste trabalho para obter algumas matrizes de quantização e realizar testes com essas matrizes foram imagens de ressonância magnética e ultra-sonografia, obtidas de várias posições do corpo do paciente, tais como: sagital, axial e coronal. As imagens de ultra-sonografia utilizadas foram: do fígado, do rim, da bexiga e do útero, do coração e da próstata. E as de ressonância magnética utilizadas foram: do abdome, do encéfalo, do joelho, da coluna lombar e do tornozelo. Nesta dissertação não são utilizados os nomes técnicos de todas essas imagens e não são informadas as posições em que foram feitos os cortes. Foi feita uma simplificação nomeando as imagens com nomes simples como pode ser observado nas Figuras 3.1 a 3.13.

O tamanho da maioria das imagens utilizadas neste trabalho é 256 x 256 pixels, mas existem algumas imagens com 512 x 512 pixels. As imagens de ultra-sonografia foram gravadas com 256 níveis de cinza no formato bitmap e as imagens de ressonância magnética foram gravadas com 2048 níveis de cinza no formato DICOM. Nesta dissertação foi feito uma quantização nessas últimas imagens para que elas ficassem com 256 níveis de cinza. Pois, como citado por Gonzalez [15], o ser humano só consegue perceber duas dúzias de níveis de cinza.

Na obtenção das matrizes de quantização deste trabalho as imagens foram selecionadas de forma a considerar amostras que representassem a diversidade delas, seja pela seleção visual ou pela seleção aleatória realizada por um programa de computador.

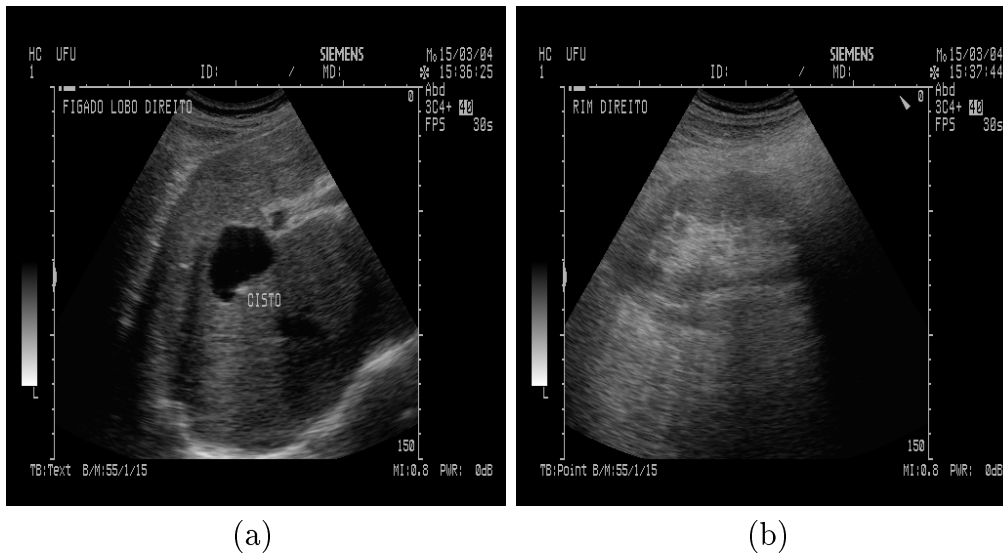


Figura 3.1: (a) Imagem do fígado; (b) imagem do rim.

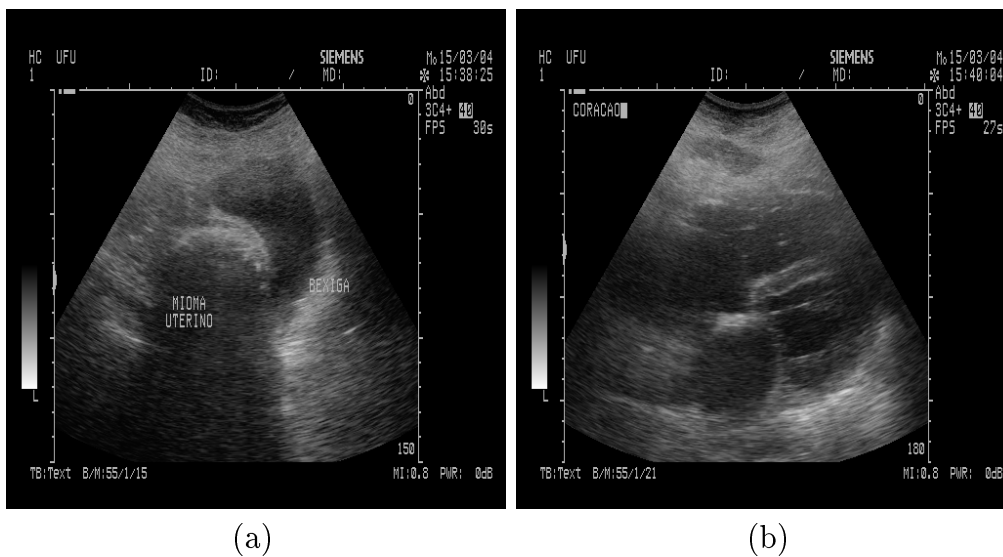


Figura 3.2: (a) Imagem da bexiga e do útero; (b) imagem do coração.



Figura 3.3: Imagem da próstata.

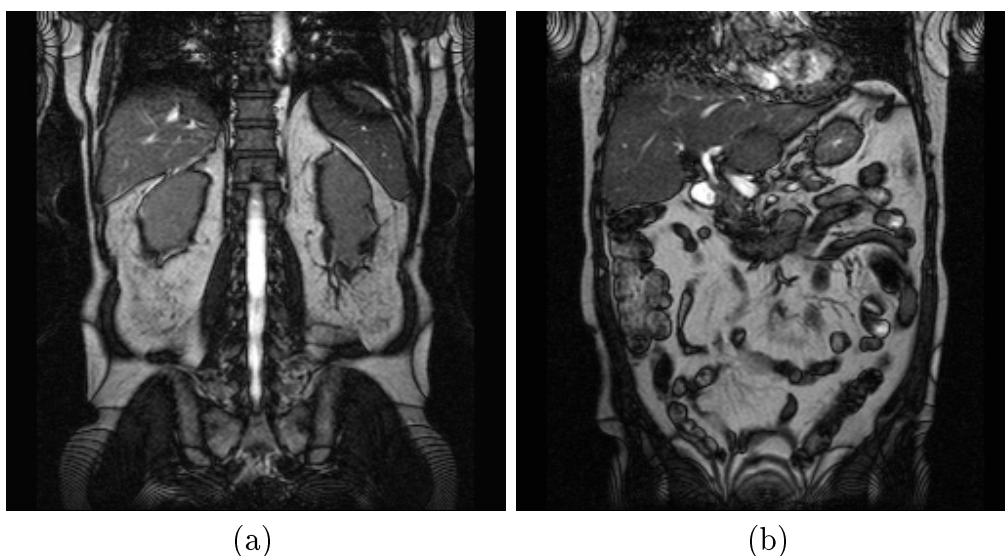


Figura 3.4: Imagens do abdome.

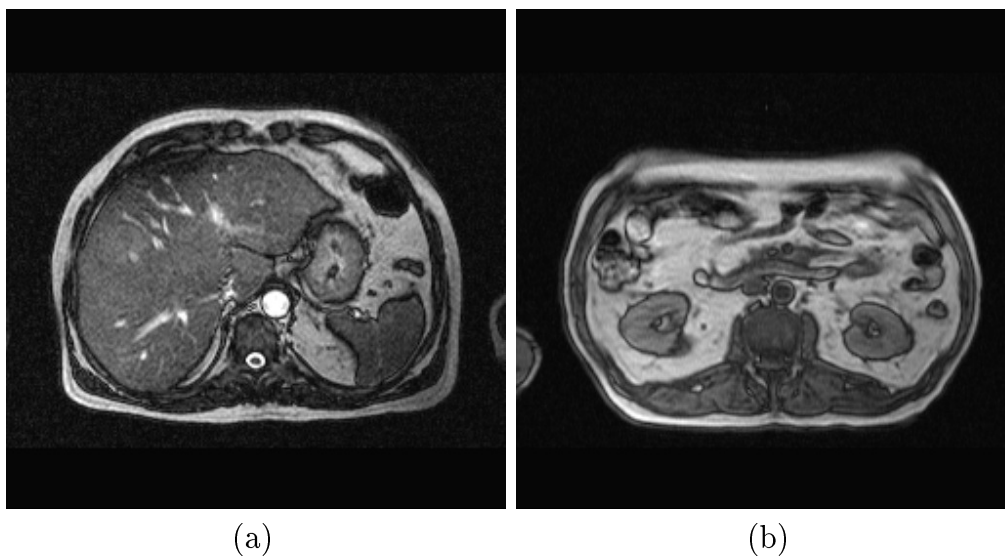


Figura 3.5: Imagens do abdome.

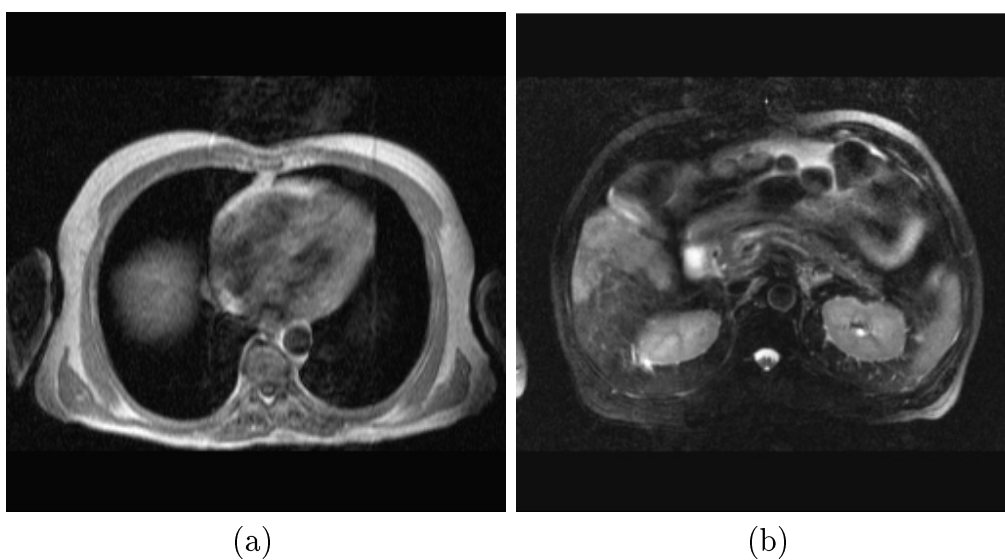


Figura 3.6: Imagens do abdome.

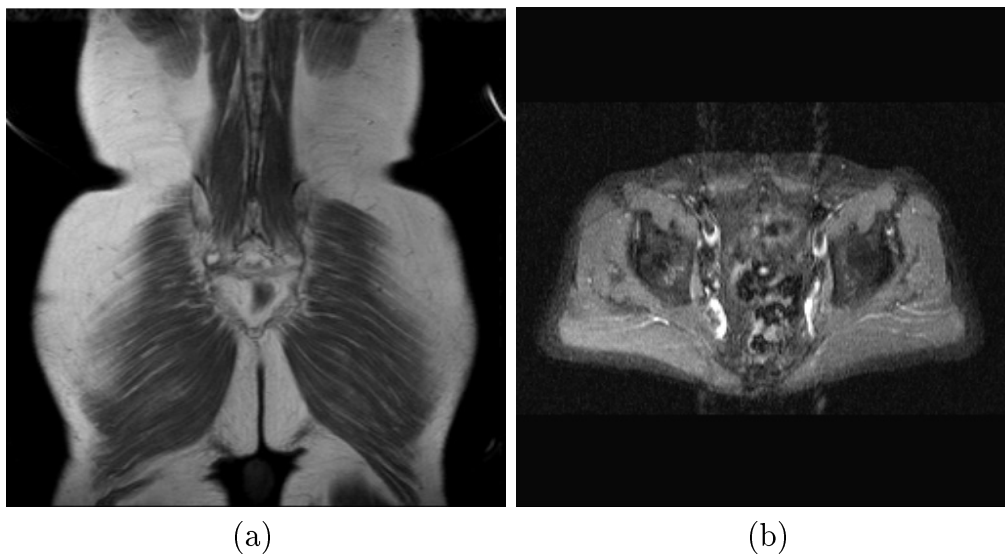


Figura 3.7: Imagens do abdome.

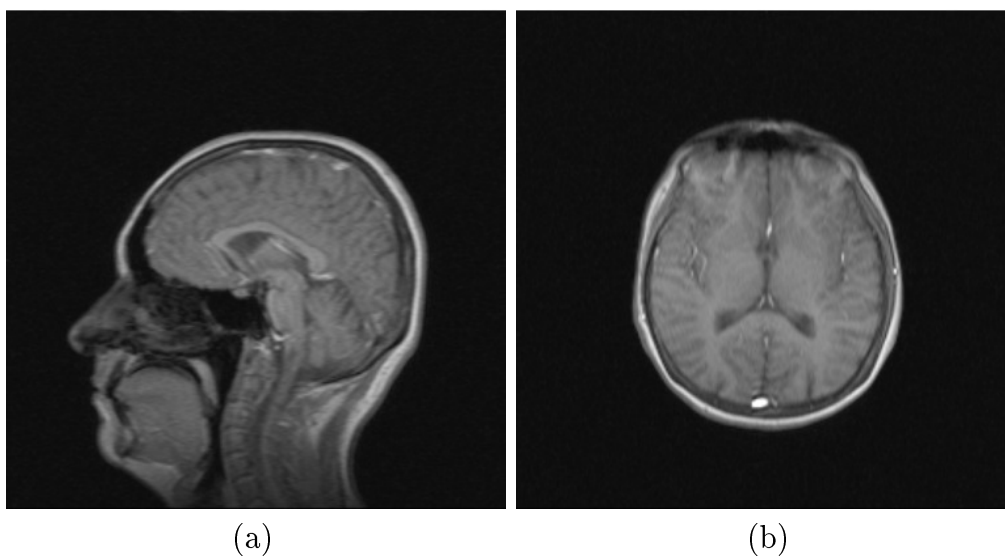


Figura 3.8: Imagens do encéfalo.