

Region of interest and windowing based progressive medical image delivery using JPEG 2000

Nithin Nagaraj^a, Sudipta Mukhopadhyay^b, Frederick W. Wheeler^c, Ricardo S. Avila^d

^{a,b}GE Global Research, John F. Welch Technology Center, Bangalore 560066, India.

^{c,d}GE Global Research, Niskayuna, NY 12309, USA.

ABSTRACT

An important telemedicine application is the perusal of CT scans (digital format) from a central server housed in a healthcare enterprise across a bandwidth constrained network by radiologists situated at remote locations for medical diagnostic purposes. It is generally expected that a viewing station respond to an image request by displaying the image within 1-2 seconds. Owing to limited bandwidth, it may not be possible to deliver the complete image in such a short period of time with traditional techniques. In this paper, we investigate progressive image delivery solutions by using JPEG 2000. An estimate of the time taken in different network bandwidths is performed to compare their relative merits. We further make use of the fact that most medical images are 12-16 bits, but would ultimately be converted to an 8-bit image via windowing for display on the monitor. We propose a windowing progressive RoI technique to exploit this and investigate JPEG 2000 RoI based compression after applying a favorite or a default window setting on the original image. Subsequent requests for different RoIs and window settings would then be processed at the server. For the windowing progressive RoI mode, we report a 50% reduction in transmission time.

Keywords: JPEG 2000, teleradiology, window-level, windowing, region of interest, progressive transmission, lossless compression, compression.

1. INTRODUCTION

The advent of 3-dimensional image processing using digital computers and CT and MR has greatly increased the amount of data to be handled, processed, stored and retrieved. In 1985, the first clinical network was implemented to allow digital diagnostic images to be shared between physicians via computer network, allowing a doctor in Boston to review a CT examination from a patient in Beijing, China. This opened the floodgates and hospitals and clinical environments started moving towards digitization, processing, storage and transmission of medical images, and the trend in healthcare information technology has been increasingly multimedia oriented². The basic motivation is to represent medical images in a digital format to support image transfer and archiving, and the manipulation of visual diagnostic information in new and useful ways, such as image enhancement and volume rendering. The Picture Archiving and Communications Systems (PACS) community envisions a completely digital environment in hospitals for the acquisition, storage, communication and display of large volumes of images of various modalities³.

To be comparable with current analog film-based medical images, digitized images must be of high quality, requiring high-resolution and, therefore, having a large amount of data in general. To maintain such large medical images with the smallest possible number of bits, data compression is essential and plays a very important role in minimizing storage requirements and speeding transmission across low bandwidth channels. The primary goal of medical image compression must be to achieve the best possible fidelity for the available communication and storage channels. Current research objectives are to identify robust compression and storage solutions for digital medical image, and to find suitable image quality measurement methods to compare efficiency and performance of different compression methods¹.

JPEG 2000⁵ is the new state-of-the-art image compression standard designed for broad range of applications,

Send correspondence to Nithin Nagaraj. Email: nithin.nagaraj@geind.ge.com.

including the compression and transmission of medical images, which is central to our discussion here. The new standard is based on wavelet technology and a layered file format that offers lossless compression, diagnostic-quality lossy compression and advanced system level functionality. JPEG 2000 was created by the Joint Photographic Experts Group (“JPEG”) under aegis of the ISO and ITU. Over 320 organizations from 21 countries contributed their expertise to the new standard, which was formally approved in January 2001 (Part 1)⁴. JPEG 2000 has been selected for inclusion in the DICOM standard for medical image transfer (DICOM Working Group 4 – Image Compression Group). DICOM Supplement 61 was ratified in November 2001 adding JPEG 2000 Transfer Syntaxes to the protocol. Of recent, there has been significant interest in telemedicine and teleradiology based applications and JPEG 2000⁷ seems to be a natural choice owing to its bitstream organization features such as RoI based coding and progressive transmission, which reduce waiting time for image transmission and display. With progressive transmission, a single compressed file caters to all bit rates (lossy to lossless) and enables the user to view a reduced fidelity image as it refines progressively to display a full fidelity lossless image.

In this paper, we discuss progressive image delivery solutions using JPEG 2000. Section 2 describes the two RoI coding methods of JPEG 2000, highlighting their pros and cons. In the next section, we discuss the benefit of windowing for image compression. We report extensive experimental results in Section 4 and also discuss the relative merits of the various schemes for progressive image delivery. Section 5 summarizes our key findings and hints towards a RoI and windowing based progressive image delivery using JPEG 2000, which we refer to as *windowing progressive RoI* method.

2. JPEG 2000 - ROI CODING

JPEG 2000 allows two different modes of RoI⁶ encoding. With dynamic mode, the client using the image can request more data and thus better fidelity for an RoI that the client selects. With static mode, the RoI is fixed and cannot be changed without recoding the image.

A. DYNAMIC (SEQUENCE BASED) MODE

In this mode, compressed data for wavelet coefficients affecting pixels within a requested RoI is extracted from the primary bitstream and recoded in a custom bitstream sequence. This allows random access to the RoI without fully decoding. The bit rate for the RoI and the non-RoI portions of the image can be specified exactly. Dynamic mode is particularly useful for interactive client/server applications⁷ where the RoI geometry and position are not known until the transmission has begun. Hence this is called the dynamic, or client-side, RoI mode.

B. STATIC MODE

With static, or server-side mode, the RoI must be specified prior to compression of the image. This is done by temporarily scaling up the wavelet coefficients that affect pixels within the RoI so that they appear more important to the bitplane entropy coder. Thus, a higher fidelity for the RoI relative to the background can be obtained. There are two methods in this mode namely ‘maxshift’ and the ‘scaling-based’ method. These are explained below. The actual geometry of the RoI need not be sent for the maxshift mode, instead it can be unambiguously inferred from the coefficient magnitudes after the shifting. Unlike dynamic mode, the exact rate for the RoI vs. background cannot be specified.

MAXSHIFT METHOD

In this method, the transformed coefficients of the RoI are shifted such that the least RoI coefficient is greater than the largest (maximum) non-RoI coefficient (hence the name maxshift). Some problems associated with this method are as follows:

1. It lacks the flexibility to allow an arbitrary scaling value to define the relative importance of the RoI and the background (BG) wavelet coefficients.
2. Overflow problem: Since the image samples have 12 significant bits, the wavelet coefficients are typically scaled up and the dynamic range may go up to 19 bits. This means that a shift of 19 is required for the RoI portion to be made distinct from the non-RoI region. If the implementation precision is only 32 bits, then

we will lose some of the least significant bits (LSB) of the non-RoI portion. This results in some loss in the non-RoI portion. This can be avoided by the scaling-based method.

SCALING-BASED METHOD

This method allows the use of arbitrary scaling factor unlike the maxshift method. The RoI coefficients are scaled by the chosen scaling factor and coded.

For the application alluded earlier, the static mode seems to be ideal, since we assume that we completely know the RoI geometry before transmission. Further, since we are considering medical image transmission, any loss of information in the non-RoI regions will not be tolerated. This leaves us with scaling-based mode to be the only reasonable static mode choice.

3. WINDOWING

Most medical images use 8-12 bits to specify the intensity of a pixel. In order to display on a 256 gray level (8 bit) monitor, we need to map the values from 12 bits to 8 bits. Choosing a window and a level does this. The level is the value around which the window is centered. All values within the window are mapped linearly to the range 0 – 255. The following graph illustrates this process.

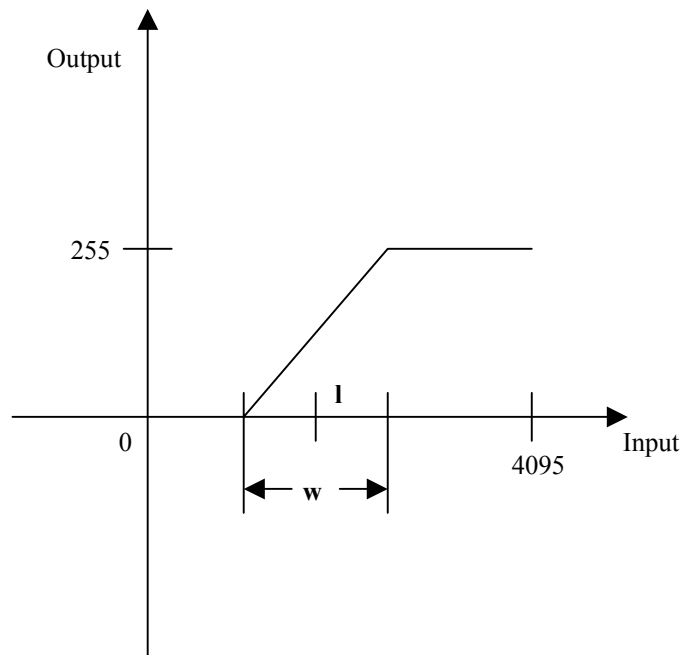


Figure 1.0. Windowing transfer function.

Here, the level “ l ” and the window “ w ” determine how intensity values of the input are scaled so that the output intensity values lie in the range $[0 - 255]$. Ideally, both the level and the window are flexible and at the discretion of the user. In this paper, we are interested in the advanced application of the windowing for compression purposes. Losslessly decoding all the bits of the wavelet coefficients of the RoI and then performing a windowing operation at the client’s end to display 8 bits of information will not yield the best compression efficiency. If we apply the windowing operation on the entire input image before compression, without selecting any region, we are effectively reducing the dynamic range of the input data set from $[0 - 4095]$ to $[0 - 255]$. This in turn reduces the number of bits required for representing the image since $[0 - 255]$ can be represented by 8 bits. Thus, the bit-depth of the image is reduced from 12-

bit to 8-bit. This new windowed 8 bit image can then be compressed to get a bitstream with a higher compression ratio. Windowing directly aids compression and we shall show by means of experiments as to how much this translates to in terms of network transmission time. Figure 4.0. shows the windowed images pertaining to two window settings for ‘Lungdemo’ (12 bit, 512x512, unsigned Lung CT) image.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Five images (lung CT digital images, Figure 3.0) with dimensions of 512 x 512 and 12 significant bits are used in these tests. The area of the lung region is around 40-55% of the total area of the image. The RoI geometries (Figure 3.1) chosen are 2 overlapping circles (for three of the images), 2 overlapping rectangles (for one image) and 2 non-overlapping rectangles (for one image), all of these located on and around the lungs. Two window settings are chosen. The first window setting (window = 450, level = 225) pertains to the lung tissue intensity. The other window setting (window = 4096, level = 2048) is a linear mapping from the whole range of 12-bit values to 8-bit values. JPEG 2000 Verification Model 8.6 was used for the study. The time taken for the transmission of the data depending on the RoI for various channel bandwidths is estimated. Here we assume that 65% of the bandwidth is available (useful) and that the rest 35% is lost in overhead, network errors etc. Both the maxshift and scaling based static modes of JPEG 2000 are used for the experiments.

To make the comparisons understandable, we depict the various possibilities for compression and transmission in the form of a tree in Figure 2.0. In this paper, results for methods (1) through (5) are presented. Method (1) is the baseline to which we compare the other cases. The most trivial case of transmitting the entire image without applying any compression whatsoever is not considered. Note that the control of loss (6) is not performed because ‘lossy’ compression³ is generally not acceptable. It would be interesting to consider trade-offs between varying degrees of compression and loss (quality) outside the RoI but that is outside the scope of this paper.

Results and discussions pertaining to these experiments follow.

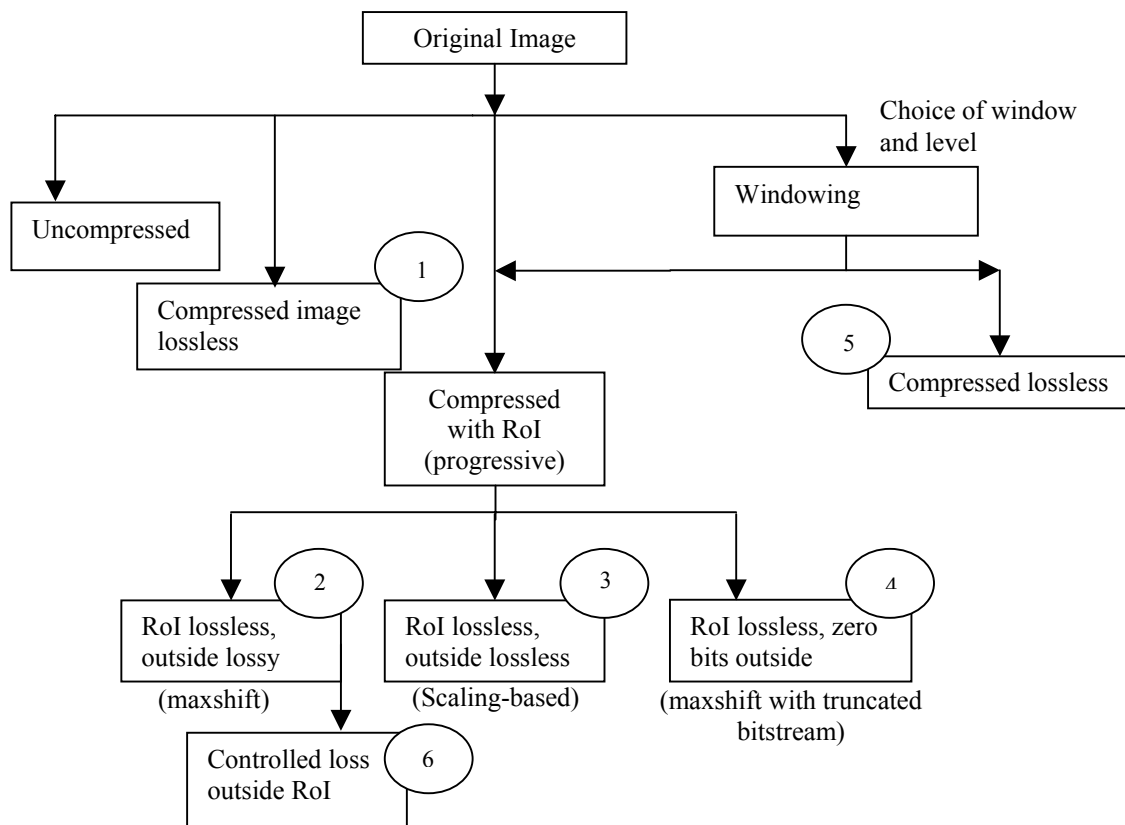


Figure 2.0. Tree depicting the various options of image compression for transmission.

EFFECT OF THE ROI ON COMPRESSION RATIO AND TRANSMISSION TIME

Table 1.0 shows the effect of selecting the RoI on compression ratio for the test images.

Image	RoI portion (%)	(A) Compressed lossless full image (bytes)	Lossless RoI, Lossless outside (bytes)	(B) Lossless RoI, (bytes)	B/A (%)	(C) Lossless RoI, Lossy outside (bytes)	(D) Lossless RoI, zero bits outside (bytes)	D/A (%)
Lungdemo	37.42	148,987	157,340	94,592	63.49	77,398	45,887	30.80
Siegel37	52.73	166,820	177,861	145,881	87.45	117,132	96,008	57.55
Brasel200	45.95	154,712	164,705	131,895	85.25	98,003	76,233	49.27
Siegel44	50.40	162,141	171,614	129,434	79.83	108,497	78,345	48.32
Brasel160	56.90	161,905	172,218	140,947	87.06	122,977	99,220	61.28

Table 1.0. Effect on compression ratio of choosing the RoI.

The following observations can be readily inferred:

1. The RoI coded bitstream is *progressive* i.e. the RoI information is placed at the beginning of the bitstream followed by the information corresponding non-RoI.
2. In Table 1.0, the method 'lossless RoI, lossless outside' takes more number of bits than the 'lossless entire image' but the RoI itself takes less number of bits (around 60-90% of the bitstream) and is available at the beginning of the bitstream (progressive), so that the RoI can be viewed sooner.
3. The method 'lossless in RoI, lossy outside' (maxshift method) takes fewer bits than the 'lossless entire image' and the RoI in this case takes around 30-60%. In other words, by this method, we would have a savings of 30-60% over the case where no RoI were chosen. This method is also progressive. Figure 4.1 shows the lossy reconstructed image and the difference image for 'Lungdemo'. The RoI portions are rendered losslessly.

Table 1.1 below shows the time taken for transmission of the compressed bit-stream through channels of differing bandwidth and for various cases for one of the test images 'Lungdemo'. Note that, similar analysis can be extended to other test images.

		Original image	Lossless compressed image	(Progressive) lossless RoI, lossy outside	(Progressive) lossless RoI, lossless outside	(Progressive) lossless RoI, zero bit outside
In kilo bytes --->		524.288	148.987	77.398	157.34	45.887
Bandwidth BW (kbps)	Rate = (BW x 0.65) / 8	Time (secs)	Time (secs)	Time (secs)	Time (secs)	Time (secs)
56	4.55	115.23	32.74	17.01	34.58	10.09
128	10.4	50.41	14.33	7.44	15.13	4.41
400	32.5	16.13	4.58	2.38	4.84	1.41
1000	81.25	6.45	1.83	0.95	1.94	0.56
10000	812.5	0.65	0.18	0.1	0.19	0.06
100000	8125	0.06	0.02	0.01	0.02	0.01

Table 1.1. Effect on network transmission times of choosing the RoI for 'Lungdemo'.

We make two important observations from Table 1.1: 1) Transmitting only the RoI with zero bits outside takes the least time and 2) As the bandwidth increases, choosing an RoI has no distinct advantage.

EFFECT OF WINDOWING ON COMPRESSION RATIO AND TRANSMISSION TIME

Table 2.0 shows the effect of windowing on compression ratio for the test images. These indicate that the gain in terms of compression as a result of windowing is more than 50%. It can be inferred from Table 2.1 that the transmission time for sending a windowed image (compressed lossless) is 50% less than that required for sending the entire original image (compressed lossless).

Image	Original file-size (bytes)	(A) Compressed file-size without windowing (in bytes)	File-size after windowing (bytes)	(B) Compressed file-size of windowed image (W1: window = 4096, level = 2048) in bytes	B/A (%)	(C) Compressed file-size of windowed image (W2: window = 450, level = 225) in bytes	C/A (%)
Lungdemo	524,288	148,987	262,144	57,070	38.31	73,176	49.12
Siegel37	524,288	166,820	262,144	67,587	40.51	69,338	41.56
Brasel200	524,288	154,712	262,144	62,282	40.26	76,615	49.52
Siegel44	524,288	162,141	262,144	64,238	39.62	65,707	40.52
Brasel160	524,288	161,905	262,144	65,474	40.44	83,761	51.73

Table 2.0. Effect on compression ratio of windowing.

		Original image	Lossless compressed image	Windowed image lossless (window = 4096, level = 2048)	Windowed image lossless (window = 450, level = 225)
In kilo bytes --->		524.29	148.987	73.176	57.07
Bandwidth BW (kbps)	Rate = (BW x 0.65) / 8	Time (secs)	Time (secs)	Time (secs)	Time (secs)
56	4.55	115.23	32.74	16.08	12.54
128	10.4	50.41	14.33	7.04	5.49
400	32.5	16.13	4.58	2.25	1.76
1000	81.25	6.45	1.83	0.9	0.7
10000	812.5	0.65	0.18	0.09	0.07
100000	8125	0.06	0.02	0.01	0.01

Table 2.1. Effect on network transmission times of windowing for 'Lungdemo'.

EFFECT OF WINDOWING + ROI ON COMPRESSION RATIO AND TRANSMISSION TIME

We now explore the effect of first windowing the test images and then compressing by choosing the RoI. Table 3.0 and Table 3.1 show the combined effect on compression ratio and network transmission time of first windowing the image and then choosing the RoI on the windowed image for compression.

Image	Compressed lossless full image without RoI & without windowing (bytes)	Compressed file size of windowed image without the RoI (bytes)	Compressed file size of windowed image with the RoI (bytes)	RoI (bytes)	% of RoI in image	% of compressed RoI w.r.t compressed full image (lossless)
Lungdemo	148,987	73,176	76,457	60,134	23.36	40.36
Siegel37	166,820	69,338	73,381	67,215	24.56	40.29
Brasel200	154,712	76,615	80,639	64,313	30.74	41.57
Siegel44	162,141	65,707	68,709	54,865	27.10	33.84
Brasel160	161,905	83,761	87,637	58,280	33.42	36.00

Table 3.0. Combined effect of windowing followed by choosing the RoI on compression ratio.

AVERAGE		Lossless compressed image	WINDOW 1 (W1)			WINDOW 2 (W2)		
Average for all images			Windowed image lossless	Windowed image lossless with the RoI	RoI only	Windowed image lossless	Windowed image lossless with the RoI	RoI only
In kilo bytes --->		158.913	73.72	77.36	49.31	63.33	69.79	56.94
Bandwidth BW (kbps)	Rate = (BW x 0.65) / 8	Time (secs)	Time (secs)	Time (secs)	Time (secs)	Time (secs)	Time (secs)	Time (secs)
56	4.55	34.93	16.2	17.07	13.13	13.92	15.26	12.51
128	10.4	15.28	7.09	7.47	5.75	6.09	6.68	5.47
400	32.5	4.89	2.27	2.39	1.84	1.95	2.14	1.75
1000	81.25	1.96	0.91	0.96	0.74	0.78	0.85	0.70
10000	812.5	0.20	0.09	0.10	0.07	0.08	0.09	0.07
100000	8125	0.02	0.01	0.01	0.01	0.01	0.01	0.01

Table 3.1. Combined effect of windowing followed by choosing the RoI on network transmission times for 'Lungdemo'.

We make the following inferences from Tables 3.0 and 3.1:

1. Selection of the RoI in a windowed image further increases the compression by about 20-30%.
2. The network transmission time is also further reduced by 20- 35%.
3. The bitstream is progressive, i.e. the RoI is received before any of the background bits.
4. Sending only the RoI after applying the windowing gives the least transmission time of all the schemes.
5. Different window settings perform differently with the RoI but this difference is negligible.

5. CONCLUSIONS

We summarize our findings as follows:

1. Selection of the RoI, in general, improves the compression ratio and reduces the number of bytes required by a ratio approximately equal to the portion of the image taken up by the RoI. We have the option of making the non-RoI part lossy or lossless by choosing either the maxshift or the scaling-based mode respectively.
2. Windowing alone yields a significant gain in compression ratio and reduction in transmission time.
3. Windowing the image followed by the RoI coding results in more than 50% reduction in transmission time compared to sending the entire image compressed losslessly.

We list several methods of image transmission in order of the time it would take from the slowest to the fastest:

- 1) Uncompressed.
- 2) Lossless RoI, lossless outside.
- 3) Lossless entire image (without RoI).
- 4) Lossless in RoI, lossy non-RoI
- 5) Lossless in RoI, zero bits outside.

Among methods, which involve windowing, we have the following list in order of time it would take from the slowest to the fastest:

- 1) Uncompressed.
- 2) Windowed and compressed lossless with RoI.
- 3) Windowed and compressed lossless without RoI.
- 4) Windowed, compressed lossless with RoI, zero bits in non-RoI.

In Figure 5.0, we hint diagram a system that exploits the observations made above. We refer to this method as *windowing progressive RoI* owing to the fact that the original image is first windowed and the RoI coding is performed on the windowed image. Progressive transmission means that the RoI portion is received before the rest of the image. Note that by choosing the scaling-based mode, we can render both the RoI and non-RoI in a lossless fashion. By using maxshift method, we will incur loss in the non-RoI portions but this would be the fastest. Figure 5.1 shows the decoding end at the client. Subsequent requests for a different RoI and window setting would then have to be processed at the server. We conclude by saying that the proposed *windowing progressive RoI* based medical image delivery using JPEG 2000 would require less transmission time (by 50%) than conventional schemes and hence result in lesser waiting time for the end radiologist, making this a strong candidate for telemedicine applications.

REFERENCES

1. J. Oh, *A Review on Medical Image Compression and Storage Options*, Educational Technology Technical Report Series, ISSN 1463-9424, The University of Birmingham, January 1999.
2. S. Wong and H. Huang, *A Hospital Integrated Framework for Multimodality Image Base Management*, IEEE Transactions on System, Man, and Cybernetics 26, No.4, pp. 455-469, July 1996.
3. S. Wong, L. Zaremba, D. Gooden, H. Huang, *Radiologic Image Compression-A Review*, Proceedings of the IEEE 83, No.2, pp. 194-218, February 1995.
4. *JPEG 2000 Image Coding System, Part1*, Final Draft International Standard Part 1 with Corrigendum 1 and Draft Corrigendum 2, ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8), 5 July 2001.
5. D. Taubman and M. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards, and Practice*, Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Boston, November 2001.
6. C. Christopoulos, J. Askelof and M. Larsson, *Efficient methods for Encoding Regions of Interest in the Upcoming JPEG 2000 Still Image Coding Standard*, IEEE Signal Processing Letters, Vol. 7, No. 9, pp. 247-249, September 2000.
7. D. S. Cruz, T. Ebrahimi, M. Larson, J. Askelof, and C. Christopoulos, *Region of Interest Coding in JPEG 2000 for Interactive Client/Server Applications*, IEEE Third Workshop on Multimedia Signal Processing, pp. 389-394, September 1999.

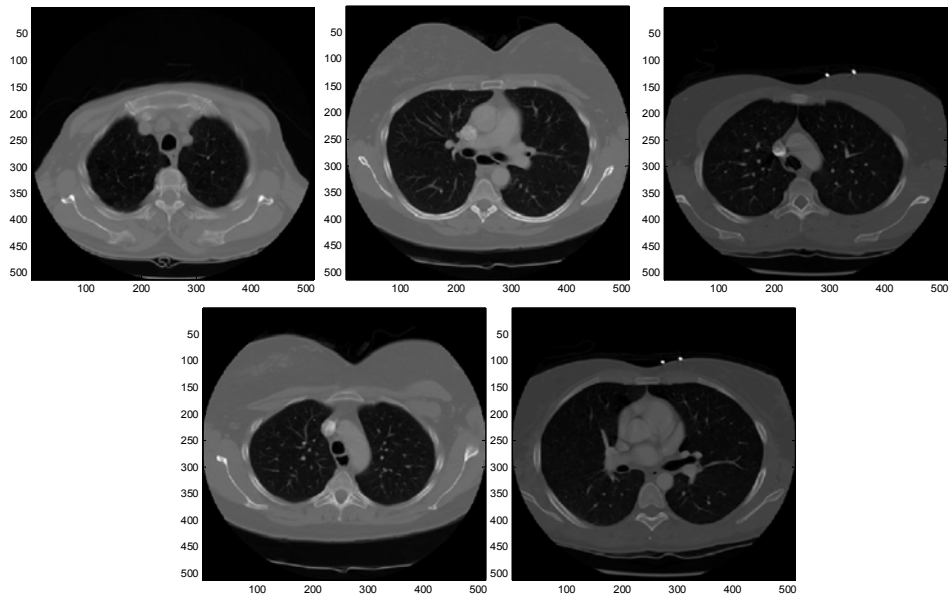


Figure 3.0. The test images 'Lungdemo', 'Siegel37', 'Brasel200', 'Siegel44', 'Brasel160' used for the experiments. (order: upper left, upper center, upper right, lower left, lower right)

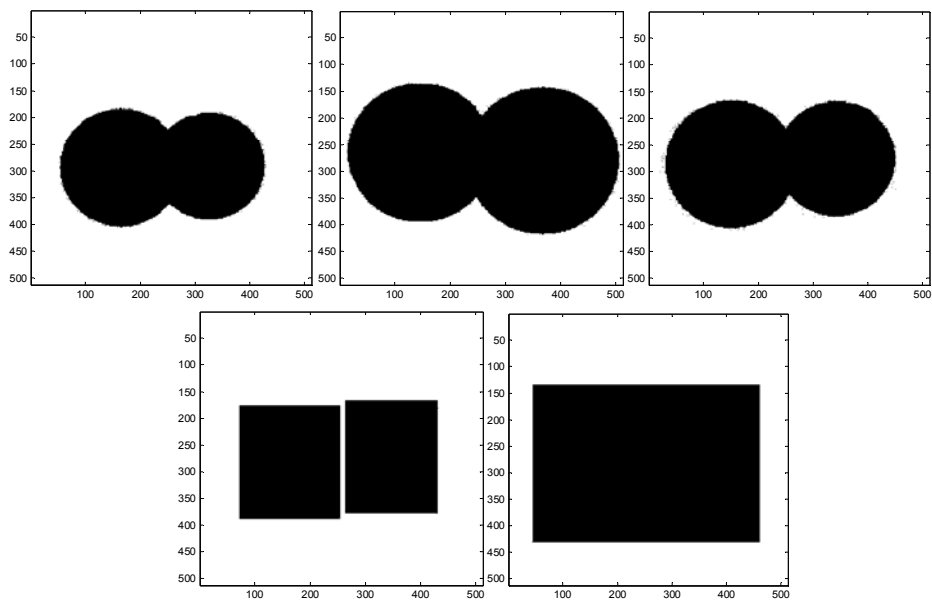


Figure 3.1. The ROI masks used for the test images.

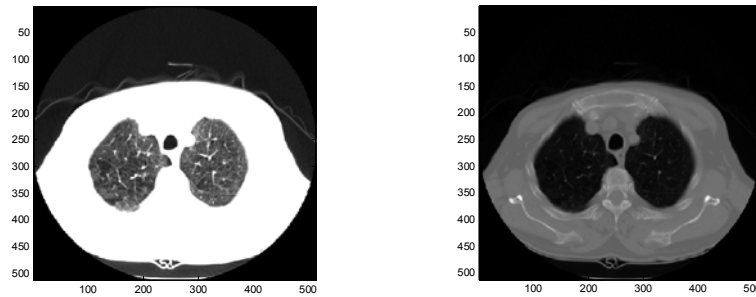


Figure 4.0. The windowed images of 'Lungdemo.'
 Left: Window = 450, Level=225, Right: Window =4096, Level = 2048.

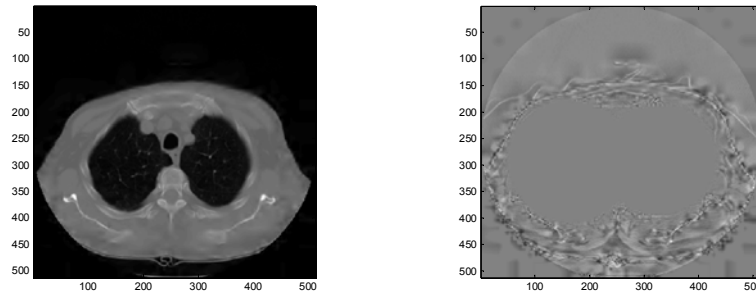


Figure 4.1. Example of 'maxshift' method of ROI coding ('Lungdemo').
 Left: Reconstructed image (the ROI is lossless, lossy outside), Right: Difference image showing the loss.

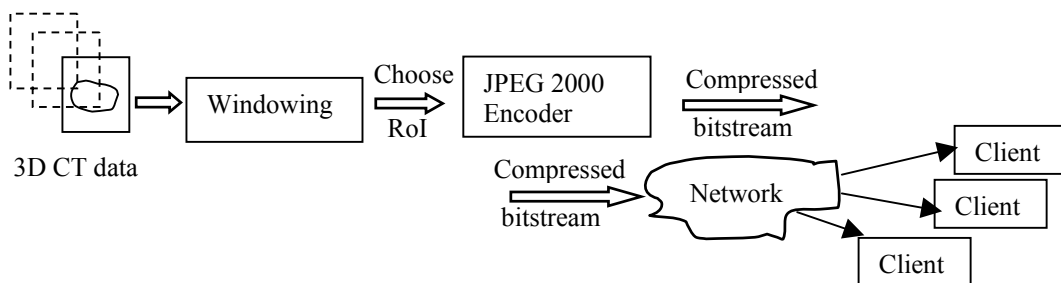


Figure 5.0. Windowing progressive ROI: Windowing followed by ROI coding using JPEG 2000.

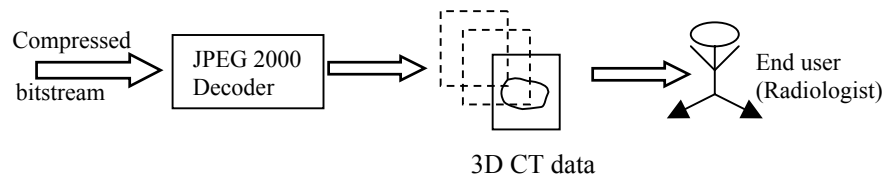


Figure 5.1. Decoding at the client end.