HIGHLY PARALLEL LINE BASED IMAGE CODE FOR MANY CORES

Dr.M.Manikandan, T.Dhanushya, C.Dhivya, P.S.Dinesh

Assistant Professor

Department of Electronics Engineering

Anna University,Chennai-600044

maniiz@annauniv.edu

ABTRACT-Computers are creating alongside another pattern from the double centre and quad-centre processors to ones with tens or even several centres. Mixed media, as a standout amongst the most vital applications in PCs, has a pressing need to configuration parallel coding calculations for pressure. Taking intraframe/picture coding as a begin point, this paper proposes an immaculate line-by-line coding plan (LBLC) to address the issue. In LBLC, an information picture is handled line by line consecutively, and each line is separated into little settled length sections. The pressure of all portions from forecast to entropy coding is totally free. Every single existing calculation broadly receives the piece based structure from expectation and change to entropy coding, where the coding of one square is reliant on the accessibility of its left, upper-left, and upper-right squares. Such a very reliant structure is not exactly appropriate for parallel coding, especially for picture and intra outline coding simultaneous at many centres. In this way, every one of the processors accessible in multi centre stage can't ready to handle parallelly. The LBLC calculation proposed in this venture is absolutely in light of line by line coding which is reasonable for the multi centre stage.

1. INTRODUCTION

The ceaseless development of mixed media and Internet applications, the necessities and prerequisites of the innovations utilized, developed and advanced. To address these necessities and prerequisites in the particular region of still picture pressure, numerous proficient procedures with impressively extraordinary elements have as of late been created for both lossy and lossless pressure. The target of picture pressure is to lessen insignificance and excess of the picture information keeping in mind the end goal to have the capacity to store or transmit information in a productive frame. The above calculations can be partitioned into two particular classes: they are either lossless or lossy. Lossless calculations don't change the substance of a document. On the off chance that you pack a document and after that decompress it, it has not changed. The accompanying calculations are lossless: Flat/collapse pressure, Huffman pressure, LZW pressure, RLE pressure. Lossy calculations accomplish better pressure proportion's by specifically disposing of a portion of the data in the document. Such calculations can be utilized for pictures or sound documents however not for content or program information. The accompanying calculations are lossy :JPEG pressure. All the above calculations widely receives the piece based structure from expectation and change to entropy coding, where the coding of one square is reliant on the accessibility of its left, upper-left, and upper-right squares. Such an exceedingly subordinate structure is not exactly reasonable for parallel coding, especially for picture and intraframe coding.

In the PC business, we are seeing an incredible change in the design of CPUs. Because of as far as possible on enhancing CPU execution through expanding its clock frequency, Moore's law is acting from another way that many handling centres are running on one chip. These days, double centre and quad-centre processors have turned out to be impressively regular in PCs, and tens or even several centres are required to develop sooner rather than later [3]. Following this trend, it is important to overhaul or create parallel programming and calculations for some centres with the goal that wide applications can profit by the proceeded with exponential development on PCs.

A multi-centre processor is a solitary registering segment with at least two autonomous genuine processors (called "centres"), which are the units that read and execute program guidelines. The guidelines are customary CPU directions, for example, include, move information, and branch, however the different centres can run various directions in the meantime, expanding general speed for programs agreeable to parallel registering. In any case, all the picture handling calculations that were - specified above can't be effectively utilized as a part of multicore stage. As they utilizes piece based coding methods in which the remaking of current square requires the reproduced past square. In this way, every one of the processors accessible in multicore stage can't ready to prepare parallel.

In piece based coding system, the whole picture is portioned into squares of settled size. At that point each square is encoded, packed and reproduced. For the reproduction of current piece, it requires the remade past square. Thus, the squares can be handled one after the main. We can't handle every one of the squares at the same time. Subsequently the square based coding is not a proficient method for multicore stage as we can't ready to prepare the pieces parallely. Along these lines, at once, we can utilize just a single centre. This is the principle disadvantage of piece based coding calculations.

There are many plans which have been proposed to parallelize the coding procedure. One proposed conspire is to arrange the forecast, change, and entropy coding of Macro Blocks (MBs) as a pipeline and dole out them to different centres for parallel figuring. This class of methodologies can accomplish constrained parallelisms if workloads are unequal at various centers. Two times speedup is accounted for superior quality arrangements on broadly useful quad-center PCs in [5].

Another proposed plot is to segment a picture into a few districts that are alluded to as cuts. The coding of cuts can be done autonomously by various centres. They give great parallelism yet would bring about a huge misfortune on coding execution if there are an excessive number of cuts (e.g., more than eight).

In spite of these endeavours on parallelizing existing coding calculations, to the best of our insight, we have not seen any endeavour on growing new coding models that can bolster full parallelism yet at the same time keep high coding productivity. In this paper, roused by the great execution of our past work on consolidating square based coding and line-based coding [5], [6], an immaculate line-by-line picture coding (LBLC) plot is proposed in this venture to oblige many-centre handling.

2. BACKGROUND STUDY

In a perfect world, a picture pressure procedure expels repetitive and unimportant data, and productively encodes what remains. Essentially, it is regularly important to discard both non excess data and pertinent data to accomplish the required pressure. In either case, the trap is discovering strategies that enable imperative data to be productively removed and spoken to.

In lossy (irreversible) pressure, the reproduced picture contains corruptions as for the first picture. Both the measurable repetition and the perceptual insignificance of picture information are misused. Significantly higher pressure proportions contrasted with lossless. Picture quality can be exchanged for pressure proportion. The term outwardly lossless is frequently used to portray lossy pressure conspires that outcome in no obvious corruption under an arrangement of assigned survey conditions

In Lossless (reversible) pressure, the picture after pressure and decompression is indistinguishable to the first. Just the factual excess is abused to accomplish pressure Data pressure systems, for example, LZW or LZ77 are utilized as a part of GIF, PNG, and TIFF document designs and the Unix "Pack" charge. Picking a Compression Algorithm relies upon the application, a few elements to consider, for example, Image quality ,Operational piece rate ,Constant piece rate , settled piece rate , consistent quantization, Computational intricacy ,System similarity and pressure standard, Input picture attributes, Data sort and past handling, Output picture applications, Spatial Accuracy.

3. EXISTING IMAGE COMPRESSION TECHNIQUES

A. DCT

A discrete cosine change (DCT) communicates a succession of limited numerous information focuses as far as an entirety of cosine capacities swaying at various frequencies. DCTs are critical to various applications in science and designing, from lossy pressure of sound (e.g. MP3) and pictures.

In figuring, is an ordinarily utilized technique for misfortune pressure for advanced photography (picture). The level of pressure can be balanced, permitting a selectable tradeoffs between capacity size and picture quality. JPEG normally accomplishes 10:1 pressure with minimal noticeable misfortune in picture quality. As the average utilization of JPEG is a lossy pressure strategy, which to some degree lessens the picture devotion, it ought not be utilized as a part of situations where the correct multiplication of the information is required, (for example, restorative and logical field). JPEG is additionally not appropriate to documents that will experience different alters, as some picture quality will normally be lost each time the picture is decompressed and recompressed, especially if the picture is edited or moved, or if encoding parameters are changed

B. HUFFMAN COMPRESSION

Otherwise called Huffman encoding, a calculation for the lossless pressure of records in light of the recurrence of event of an image in the document that is being compacted. The Huffman calculation depends on factual coding, which implies that the likelihood of an image has an immediate bearing on the length of its portrayal. The more plausible the event of an image is, the shorter will be its bit-measure portrayal.

C. RUNLENGTH CODING

Run-length encoding (RLE) is an exceptionally basic type of information pressure in which keeps running of information (that is, groupings in which similar information esteem happens in numerous sequential information components) are put away as a solitary information esteem and check, as opposed to as the first run. This is most helpful on information that contains numerous such keeps running: for instance, basic realistic pictures, for example, symbols, line drawings, and activities. It is not valuable with documents that don't have many keeps running as it could significantly expand the record measure.

4. LITERATURE SURVEY

Shekhar Borkar[1] exhibited the many-centre engineering, with hundreds to thousands of little centres, to convey exceptional process execution in a reasonable power envelope. We examine fine grain control administration, memory data transfer capacity, on pass on systems, and framework versatility for the many-centre framework.

Erik B vander Tol and Egbert G.T.Jaspers[7] proposed to segment the information over the processors, rather than customary practical dividing. Preference of this approach is the inborn territory of information, which is critical for correspondence productive programming usage. Thus, a product usage is examined, empowering e.g. SD determination H.264 deciphering with a two-processor engineering, though High-Definition (HD) unravelling can be accomplished with an eight-processor framework, executing a similar programming. Aside from the impressive change in memory data transmission, this novel idea of apportioning offers a characteristic approach for ideally adjusting the heap of all processors, along these lines additionally enhancing the general speedup.

Klaus Schoffmann and Markus Fauster[2] proposed and assess diverse methods for utilizing multithreading to accelerate our .NET actualized decoder. While cut based methodologies scale best, this is not an exible approach on account of the dependence on uniquely encoded streams. Utilitarian dividing and large scale square pipelining turn out to be a decent option for all assessed recordings.

Xiulian Peng and Jizeng Xu[5][6] proposed a line-based approach for picture coding. Not at all like regular square based picture coding, the proposed strategy takes one line as an essential unit in coding and remaking. Such a structure enables us to perform better expectations between neighbouring units and bears us greater adaptability when coding every unit. We additionally exhibit a productive forecast strategy utilizing versatile expectation channels for this structure. In view of the neighbouring data, we apply the best expectation channel on each line to get top notch forecasts. We demonstrate that the proposed line-based coding can perform relatively or shockingly better than the best in class piece based coding.In this paper we additionally enhance the line-based coding plan by misusing long-remove relationships in forecast. Line-by-line format coordinating (LTM) is acquainted with play out the long-separate expectation.

Detlev Marpe and Heiko Schwarz[4] proposed the Context-Based Adaptive Binary Arithmetic Coding (CABAC) structure additionally incorporates a novel low multifaceted nature strategy for parallel math coding and likelihood estimation that is appropriate for effective equipment and programming usage.

Limin Liu and Yuxin Liu[3] explored the utilization of straight expectation for intra piece coding and propose a setting versatile intra forecast approach. We utilize the slightest square expectation and infer the direct expectation coefficients utilizing reproduced information. The straight expectation coefficients certainly implant the nearby surface attributes and in this way the intra forecast mode is adaptively balanced by the neighbourhood setting. No additional overhead is required for flagging the coefficients since the decoder basically rehashes a similar inferring process. By turning on the new intra forecast mode, we will exhibit an enhanced coding proficiency execution. The drawback of the proposed approach is the expanded computational unpredictability at both the encoder and the decoder.

5. PROPOSED SYSTEM

A. HIGHLY PARALLEL LINE-BASED IMAGE CODING FOR MANY CORES

These days, double centre and quad-centre processors have turned out to be significantly normal in PCs, and tens or even several centres are relied upon to rise sooner rather than later [1]. The standard picture and video pressure broadly embraces the square based structure from expectation and change to entropy coding, where the coding of one piece is reliant on the accessibility of its left, upper-left, and upper-right squares. Such an exceedingly subordinate structure is not exactly appropriate for parallel coding, especially for picture and intraframe coding.

Sight and sound, as a standout amongst the most vital applications in PCs, has a pressing need to configuration parallel coding calculations for pressure. Taking intraframe/picture coding as a begin point, this venture proposes an immaculate line-by-line coding plan (LBLC) to address the issue. In LBLC, an information picture is handled line by line sequentially, and each line is partitioned into little settled length segments, which can be coded autonomously from expectation to entropy coding. By relegating portions of a line to various centers, a high and steady parallelism can be accomplished..

Despite the fact that the proposed conspire additionally receives the line-based coding structure, it is essentially unique in relation to the past endeavours regardless of from coding strategies or execution. The proposed LBLC plan can clearly accomplish a high parallelism by separating the reliance among fragments in a similar line.

The Flow outline for the whole procedure is given beneath in the figure3.1..In this section we talk about each piece of this stream diagram in detail.

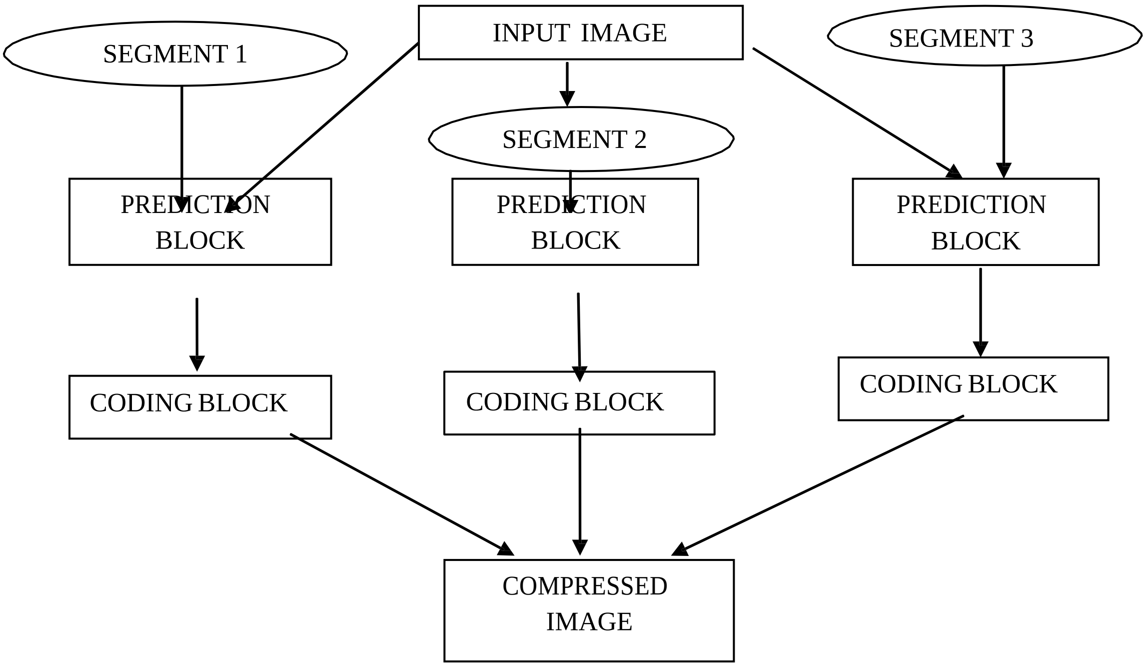


Fig 1 Flowchart of the entire LBLC scheme.

B. LBLC SCHEME

The proposed LBLC goes for a high parallelism by lessening the reliance among various centres however much as could be expected and keeping away from expansive coding execution misfortune. The essential thought of the proposed LBLC system is shown in Fig 3.1. An information M*N picture is coded line by line, i.e., from to .The line here alludes to an entire line of the picture. For each line , where , it is separated into little portions , where , with rise to lengths . The coding of these portions is totally free from one another. In this way, they can be allotted to various centres for parallel handling. After one line is coded and reproduced, it is utilized to foresee the following line. For each fragment in line , it needs the three sections , and in the past line for expectation.

The proposed LBLC plan can clearly accomplish a high parallelism by separating the reliance among sections in a similar line. The inquiry is the means by which the coding execution will be debased. There are two motivations to corrupt the coding execution. In the first place, the coding of a section can't utilize neighbouring fragments in an indistinguishable line from forecast. Contrasted and the SP (Slice Portioning) approaches, this impact is limited in the proposed line coding structure on the grounds that there are just line segments.

The pixels in a fragment are coded as a unit, where the quantity of pixels is altogether not as much as that of a MB. The coding header of a section ought to be diminished however much as could reasonably be expected. Subsequently, how to outline an effective coding plan for each line fragment turns into a key issue in the proposed LBLC. Like the structure of prescient change coding, pixels inside are first anticipated from the reproduced pixels of neighbouring lines utilizing a directional channel.

Since the relationship among neighbouring lines is considerably more grounded than that in piece based coding, the forecast channel for can be gotten from recreated pixels in past lines, bringing about both great expectation exactness and a low level of overhead. After the forecast, a 1-D change is alternatively performed on the anticipated build-ups to encourage de-correspond them. The change coefficients are then quantized and entropy-coded utilizing number juggling coding. Moreover, to adjust to various substance at the picture line, diverse units for coding, expectation, and changes are permitted, which constitute a progressive coding structure.

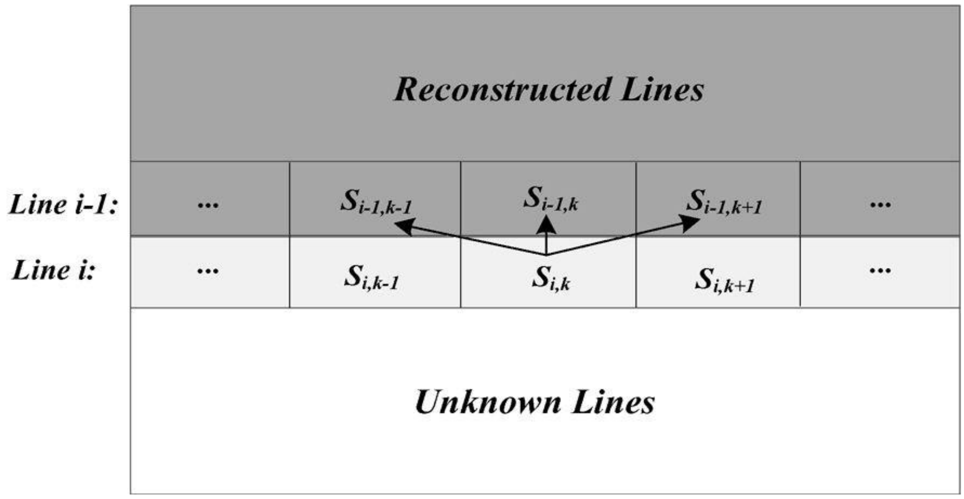


Fig 2 Proposed LBLC scheme.

C. PREDICTION FILTER BLOCK

Persuaded by our past work in [5] and [6], we proposed two verifiable expectation techniques in this venture for the forecast: by two neighbourhood expectation strategies utilizing directional channels. In the nearby forecast, for each portion to be anticipated, the neighbouring reproduced pixels in past lines constitute the preparation window appeared in Figure 3.3. The forecast parameters are figured totally from pixels inside. In this manner, there is no compelling reason to transmit them to the decoder. We have two ways to deal with infer the forecast parameters:

- Pre Defined Filters (PDF) .
- Direction-Aided Local Training (DA-LT).

1. PREDEFINED FILTERS

The Pre Defined Filters (PDF) approach predefines a set of directional filters and selects one among them using pixels inside the window W. For each pixel in the current segment, there are seven reconstructed pixels in the previous line for its prediction Fig3.4. Following this prediction pattern, each pixel inside, which is denoted as,y_i where,$y = 0,1,\dots N_t - 1$, will have seven reference pixelsy_{ik}, where $k = 0,1,\dots6$, for its prediction. Letf_j, where,$j = 0,1,\dots N-1$, with coefficients$\{w_{jk} \backslash k = 0,1,\dots 6\}$ denote the filters we have defined.

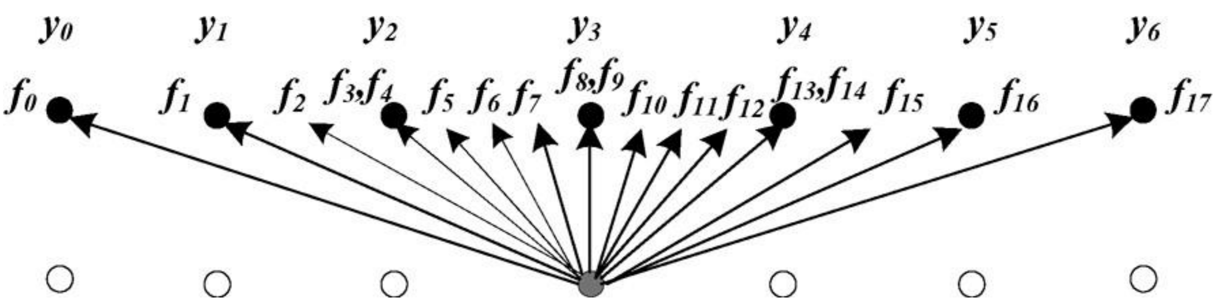


Fig 4. Predefined directional filters in the PDF. The seven pixels denoted by black circles are used to predict one pixel denoted by a gray circle.

In this strategy, as the name suggests we have an arrangement of channels. Every one of these channels have officially characterized coefficients. We need to pick the best channel among them.

Mean squared error

In insights, the mean squared error (MSE) of an estimator is one of numerous approaches to evaluate the contrast between values inferred by an estimator and the genuine estimations of the amount being estimated.MSE measures the normal of the squares of the "blunders." The blunder is the sum by which the esteem suggested by the estimator varies from the amount to be assessed. The distinction happens in light of irregularity or on the grounds that the estimator doesn't represent data that could deliver a more precise gauge.

MSE for all pixels inside the window will be given by the formula in equation 1

$$J = \arg\min_j \left(\frac{1}{N_t} \sum_{i=0}^{N_t-1} \left(y_i - \sum_{k=0}^{6} y_{ik} \cdot \omega_{jk} \right)^2 \right) \quad (1)$$

Where N_t represents the total number of predefined filters, y_i represents the current pixel to be predicted, y_{ik} represents the reconstructed pixels in the window, w_{jk} represents the filter coefficient of j^{th} filter.

1. DIRECTION-AIDED LOCAL TRAINING (DA-LT)

The second prediction method we used is DA-LT. Unlike the PDF with fixed filters, the LT approach derives the filter coefficients through an online training process, using pixels in . As depicted in Fig. 3.5, three reconstructed pixels are used to predict one pixel. All pixel sets similar to this in are used as the training data. Let denote filter coefficients. Then, the best filter for can be obtained by minimizing the prediction MSE on all pixels inside, i.e., by using the equation no (2)

$$\omega_k = \arg\min \left(\frac{1}{N_t} \sum_{i=0}^{N_t-1} \left(y_i - \sum_{k=0}^{2} y_{ik} \cdot \omega_k \right)^2 \right) \tag{2}$$

Where N_t represents the total number of predefined filters, y_i represents the current pixel to be predicted, y_{ik} represents the reconstructed pixels in the window, w_{jk} represents the filter coefficient of jth filter.

Algorithm 1: Huffman coding

- Initialize list of probabilities with the probability of each symbol
- Search list of probabilities for two smallest probabilities, pk _ and pl_.
- Add two smallest probabilities to form a new probability, pm = pk_ + pl_.
- Remove pk_ and pl_ from the list.
- Add pm to the list.
- Go to step 2 until the list only contains 1 entry

Huffman decoding

- Construct decoding tree based on encoding table
- Read coded message bit-by-bit:
- Travers the tree top to bottom accordingly.
- When a leaf is reached, a codeword was found corresponding symbol is decoded
- Repeat until the pseudo-eof symbol is reached.

The program flow of arithmetic coding can be viewed as shown in the Fig below.

To decode the message, a similar algorithm is followed, except that the final number is given, and the symbols are decoded sequentially from that. The table below presents a simple comparison between these compression methods.

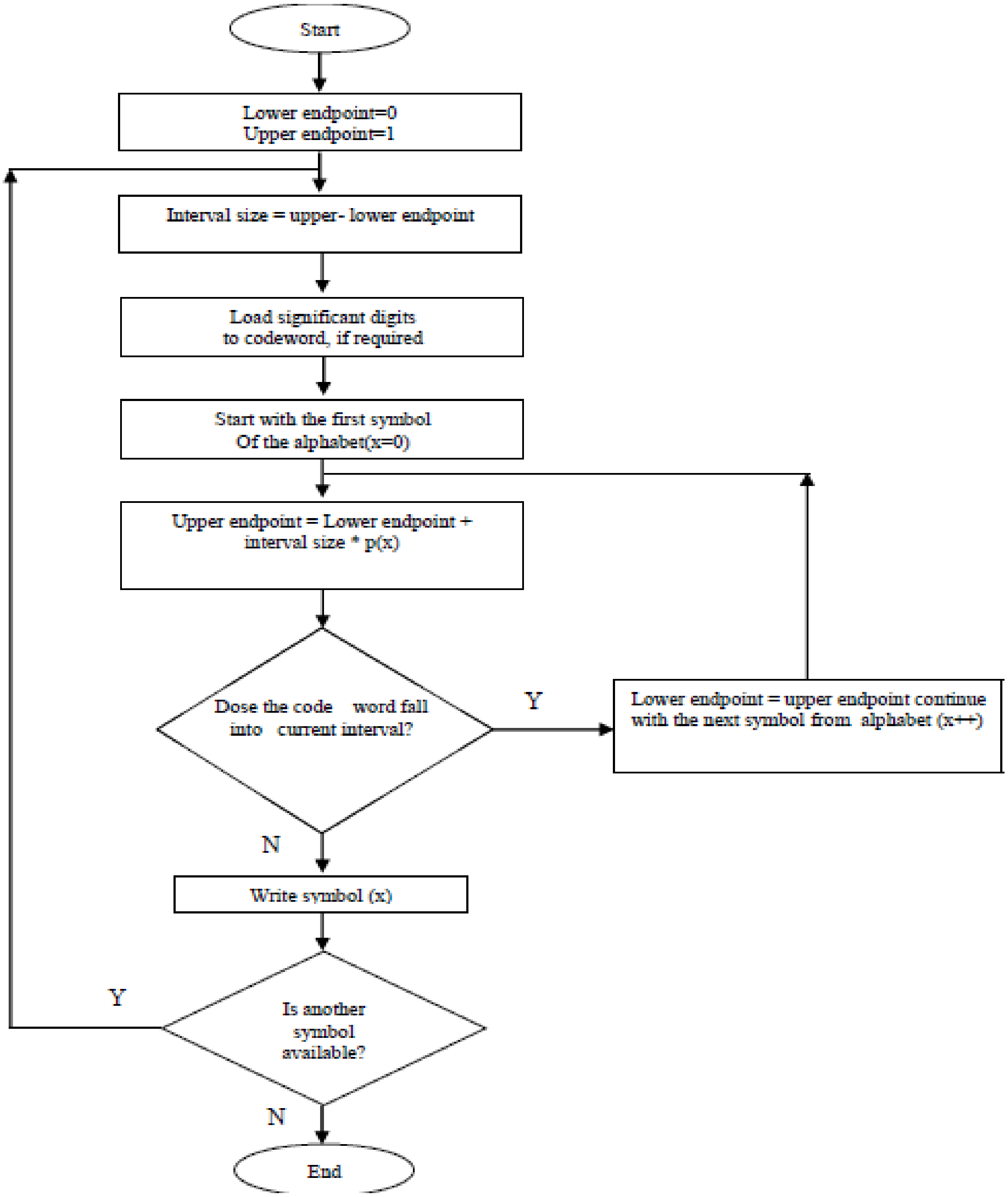
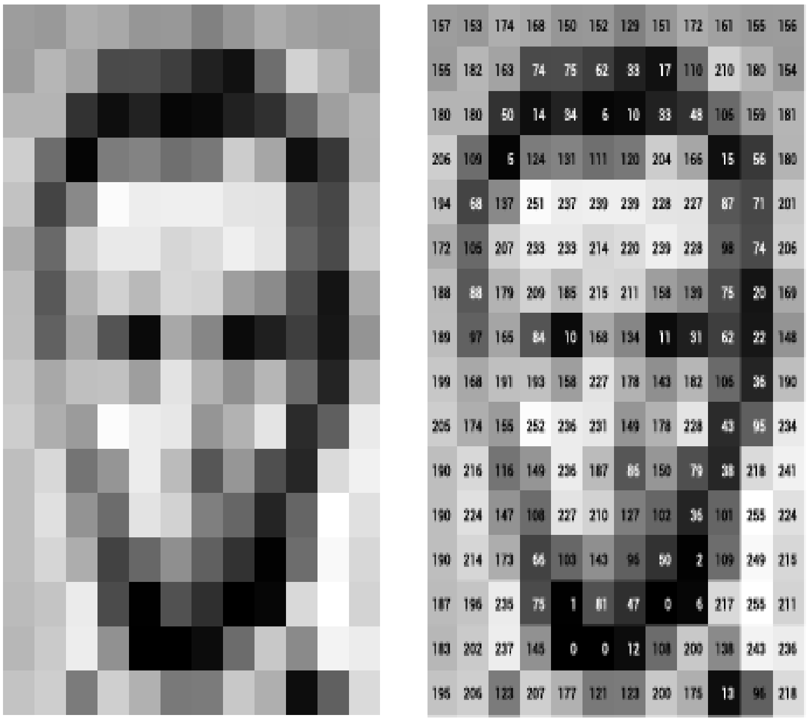


Fig.4 Flowchart of arithmetic encoding process

The smallest element of an image is called as pixel, and it can store a value proportional to the light intensity at that particular location. It is the fundamental unit of color on a computer demonstrate or in a computer image.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Fig.5 Example of pixel data diagram

Three components of the color spectrum – RGB is specifically described as a pixel. Details of image holds are called as image resolution, and it has various types, i.e. pixel resolution, temporal resolution, spatial resolution, spectral resolution. From this pixel resolution is discussed below.

Fig.6 (a) Low resolution image (b) High resolution image

A total number of pixels count in a digital image is referred as a resolution in pixel resolution. For example, if an image has P rows and N columns, then its resolution can be defined as P X N.

A large set of values to a smaller set process is called as quantization. Then the input value and quantized value difference are referred as quantization error. The matrix of picture elements or pixels is called as digital image if the quantized image differs from the unquantized image, so errors have been created. Image storage process requires a number of bits from the amount of information in the quantized image. A number of bits of data is higher than the number of bits of information. The subject of information substance is very significant in imaging.

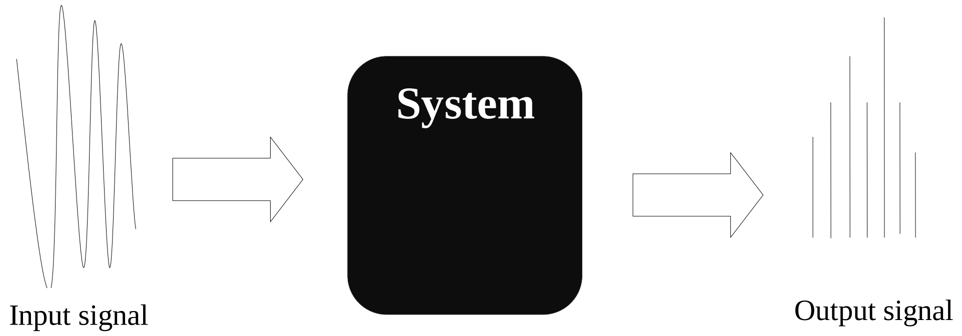


Fig.7 Quantization

Sampling is done on the X axis and quantization is done in the Y axis. So that means digitizing the gray level resolution of an image is done in quantization. The simplest type of images is called as binary images which take two values normally white and black, or 1 and 0. It will take only one binary digit to characterize every pixel, so it is referred as a 1-bit image. Using threshold operation, binary images are frequently created from the gray scale images.

For gray scale images, pixels value or average pixel value are simply used in a window of every pixel. It is a range of monochromatic shades to white from black. For gray scale images, segmentation process helps to divide an image into non-overlapping, homogenous regions having related objects. The gray level histogram of the image is bi-model. By basically selecting a threshold value, objects are clearly noticeable from the background by considering the valley among two peaks of the histogram.

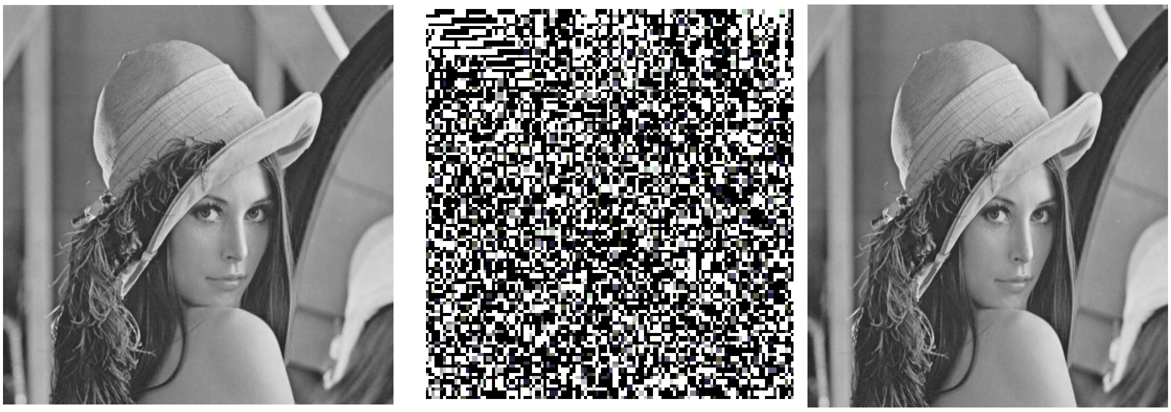
Gray level histograms of the real time images are multimodal in the real cases and precise position of distinct valleys is identified in multimodal histograms is fairly complicated. Light intensity is measured only by a grayscale. Every pixel is a scalar proportional to the brightness. Hence, black is known as the minimum brightness and the white is known as maximum brightness.

Fig.8 Examples of Binary image

Fig.9 Grayscale image of resolution 512x512

6. RESULTS AND DISCUSSION

We have applied the above algorithm for many input images. The results for various input images have been discussed in this chapter. For example consider the input image as shown in Fig 5.a



a **b** **c**

Fig 5 a. Input image, b. Compressed Image, c. Reconstructed Image

This image is then processed row by row using the LBLC algorithm proposed in this project. Each row is segmented into as many segments as number of cores available. Then each segment is passed through a single core. It is then compressed using arithmetic coding. Fig 5.b shows the compressed image of the given input image. The compression has been done using arithmetic coding.

The compressed image is then reconstructed as row by row in each core and final reconstructed image is shown in Fig 5.c

Table 2 Simulation results for various input images.

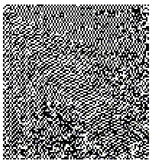
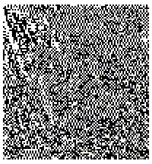
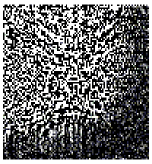
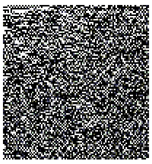
Original Image	Compressed Image	Reconstructed Image
		
		
		
		

Table 3 Comparion of results for different input images.

Test Image	Image Size	No of Cores	Compression Ratio		Time taken (secs)	
			Block based coding	LBLC	Block based coding	LBLC
Cameraman	256×256	3	16.7	3.2	23.1	21.8
Lena	512×512	3	17.2	3.6	24.6	22.0
Pirate	512×512	3	16.9	3.8	23.8	22.3
Woman	1024×1024	3	18.3	4.2	27.3	27.1

The Table 4.2 shows the comparison between the block based coding and LBLC taking compression ratio and time as the parameters. It is clearly seen time for the program to run is less in LBLC when compared to the block based coding for the same input image.The compression ratio of the LBLC clearly shows that it is an efficient algorithm for multicore processing.

7. CONCLUSION

An unadulterated line-by-line picture coding plan is proposed in this venture for some centre handling. It decreases the reliance among various centres to a base level without causing much coding execution misfortune. Taking intraframe/image coding as a begin point, this Project proposes an immaculate line-by-line coding plan (LBLC) to address the issue. In LBLC, an info picture is handled line by line consecutively, and each line is separated into little settled length sections. The pressure of all portions from forecast to entropy coding is totally free and simultaneous at many centres. The autonomous coding of each portion inside a line makes it simple to accomplish a high parallelism. Recreations performed on a genuine parallel stage demonstrates that the proposed plan can accomplish a close straight speedup as the quantity of centres increments. Notwithstanding high parallelisms, it likewise accomplishes great coding exhibitions at centre and high piece rates. These outcomes make us trust that the LBLC structure is a decent answer for exceedingly parallel coding with high coding productivity.

REFERENCES

[1] Borkar.S, “Thousand core chips: A technology perspective,” in *Proc. 44th Annu. Conf. Des. Autom.*, 2007, pp. 746–749.

[2] Klaus Schoffman. O. L., Fauste .M, and Boszormeny.L, “An evaluation of parallelization concepts for baseline-profile compliant H.264/AVC decoders,” in *Proc. Euro-Par—Parallel Processing*, 2007,Lecture Notes in Computer Science, pp. 782–791

[3] Liu.L, Liu.Y, and Delp.E.J, “Enhanced intra prediction using context- adaptive linear prediction,” in *Proc. Picture Coding Symp.*, Lisboa, Portugal, 2007.

[4] Marpe.D, Schwarz.H, and Wiegand.T, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636.

[5] Peng.X, Xu.J, and Wu.F, “Line-based image coding using adaptive prediction filters,” in *Proc. Int. Symp. Circuits Syst.*, 2010, pp. 4221–4224.

[6] Peng.X, Xu.J, and Wu.F, “Improved line-based image coding by exploiting long-distance correlations,” in *Proc. Vis. Commun. Image Process.*, 2010, pp. 77441D-1–77441D-8.

[7]Van der Tol. E.B, Jaspers .E. G. T., and Gelderblom, R. H, “Mapping of H.264 decoding on a multiprocessor architecture,” in *Proc. SPIE Conf. Image Video Commun. Process.*, 2003, pp. 707–718.