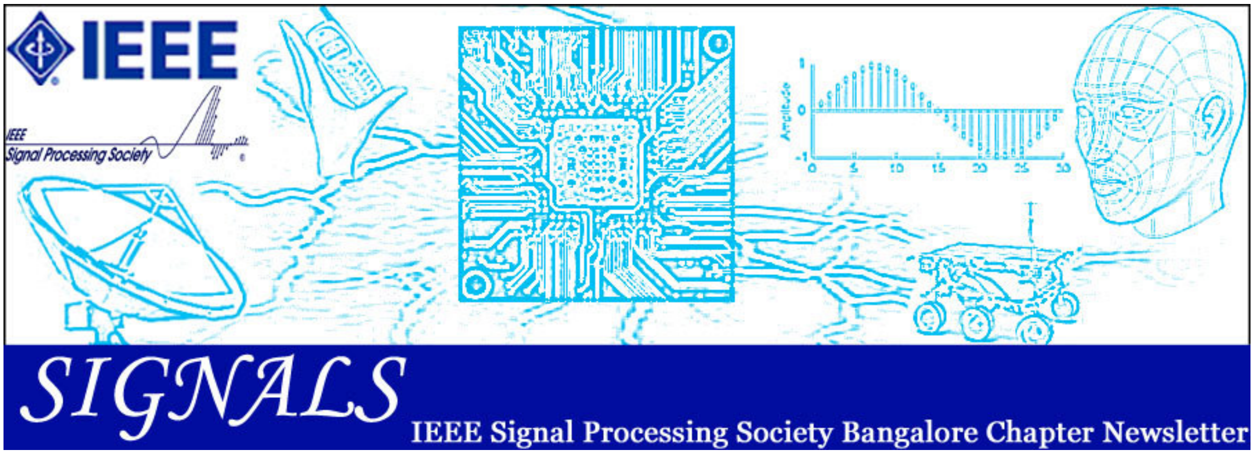

ISSUE #08, Oct 2007

FROM THE EDITORIAL TEAM

We apologize for having missed the July issue due to unforeseen circumstances.

A lot has been happening in recent months. On the communications front WiMax is going from strength to strength. ITU has recently included WiMax within the purview of IMT-2000. While skeptics point out that OFDM techniques have been seen in ITU before, most industry watchers hail this as a significant event. Cisco's acquisition of WiMax player Navini has raised WiMax's position in the mobile space a notch or two.

The video space has also seen a flurry of activity. Everyone wants faster and better quality video. Even in a traditionally low-quality domain like surveillance, video analytics is beginning to demand HD. Transcoding remains one of the holy grails of video communication. Solutions ranging from ASICS to highly programmable and even multicore processors are vying for that huge market.

Cars are becoming smart these days. They have huge computation capacity on board. This is used for diverse uses including analysis, safety, entertainment and so on, which involve significant real-time processing of sensor inputs, media and communication signals.
IEEE recently proposed to mandate IP disclosures from contributors. This is meant to allow the committees to consider the cost of a proposal while making a decision to include it in the standard. This kind of transparency is in general welcome, but the efficacy of this poliy remains to be seen.

On to the home front now. 3G wireless continues to languish in the labyrinths of regulatory and government bodies due to complications and confusion about the sharing of spectrum across

different players (CDMA vs. GSM, existing vs. new players and so on). Let us hope 3G sees the light of day in India soon! In the mean time, the first Indian Embedded Systems Conference was held recently in Bangalore. The response was massive and the quality of presentations as well as participation seems to have exceeded all expectations. A nice vignette of this ESC can be found at
http://www.mobilehandsetdesignline.com/showArticle.jhtml?articleID=202400034

Abraham Lempel has been selected for the IEEE Hamming Medal for his contributions to source coding. A well-deserved honour, in our opinion!

Finally, we are looking for good articles as usual. Please send in your ideas, abstracts to the usual email ID. Thank you.

The Editorial team

ANNOUNCEMENTS

- The recently introduced Resource Corner provides links to interesting articles that have appeared elsewhere but are worth reading. It will also include some online resources that can be every DSP engineer's friend. Do send in your feedback, suggestions and contributions to ieeesp-blr-news@ieee.org.

- To keep our newsletter vibrant, we do need articles from readers like you. Please send in a brief abstract of your article to ieeesp-blr-news@ieee.org. If you have written an article in the past which you are willing to share with this community, that is most welcome too.

MAIN COURSE

Motion Estimation for Real Time Video Encoding

Pankaj K. R. Multimedia Codecs, Texas Instruments India

ABSTRACT

The last couple of years have seen significant growth of digital video all sorts of applications. In fact, in home entertainment application traditional analogue video equipments are almost completely replaced by digital equipments. Central to this progress has been the significantly improved video coders capable of producing high quality video at considerably low bit rates. The central ideas of all these coders incorporate mechanism to reduce the temporal redundancies from frame to frame so that a frame can be represented by a small residual, and then use a spatial coding method like DCT to code the residual effectively.

The objective of this paper is to provide a tutorial overview of the temporal redundancy removal procedure. This process of reducing or removing temporal redundancy is known as the Motion Estimation (ME) and Motion Compensation (MC). In this paper we try to cover a survey of most the motion estimation process and also give comprehensive details about implementing them for real time applications. We start the paper with the basic definition of Motion Estimation and Compensation. We feel that this approach not only tells about the fundamentals of ME and MC but also provides a valuable background for beginners.

1. INTRODUCTION

A video consists of sequences of still pictures in time. When these still pictures are displayed exactly at the same rate they are captured they give the impression of a video. This principle is used for about 100 years in movie production and display. Though the same principle is applicable to digital video, some new challenges come up. Before going into digital video let's consider the case of a digital image. A digital image compression scheme like JPEG, can not be used for video just because the data rate produced by this method is still very high. This arises due to the fact that a video consists of about 30 frames (still pictures) per second. The higher bit rate arises due to the fact that in a video sequence one frame has lot of correlation with the previous frames. But unfortunately the JPEG does not exploit this redundancy. This redundancy in time domain is known as the temporal redundancy.

Motion Compensation is the process in which a portion or block of a current frame is predicted from a portion from the past frame.

Motion Estimation is process of finding the right candidate in the past frame (reference frame) for the replacement.

Motion Compensation involves only a replacement but Motion Estimation involves finding the right candidate. Therefore, motion estimation is a computationally demanding process. A better motion estimation always means less residual to code and hence means better video coding efficiency.

In this paper we devote most of the discussion towards various motion estimation approaches. We also discuss about approaches which can be used in a real time video encoder.

The paper is organized as follows. The section 2 is an introductory section primarily for the beginners. The 3rd section talks about the Motion Compensation. The section 4 is devoted towards various approaches for motion estimation. The 5th section is devoted towards the practical considerations for a real time implementation of Motion Estimation algorithm.

2. VIDEO CODING BASICS

This section describes the basic principle of video compression. Most of the video compression algorithms are lossy. The goal of a video compression algorithm is to reduce the data used to represent the original video without compromising perceptual quality of the compressed video.

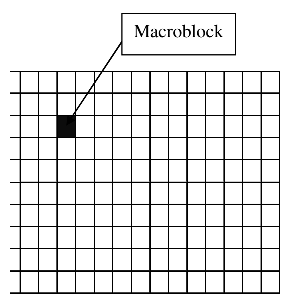

Fig 1 Video Frame and Macroblock

Video compression typically operates on rectangular-shaped groups of neighboring pixels often called a macroblock as shown in the Fig 1. Typically these macroblocks are of sizes 4X4, 8X8 and 16X16 pixels. But, there can be other sizes as well. These macroblocks are compared from one frame to the next and the video compression algorithm encodes only the differences within these blocks. This scheme works very well if there is no motion between the frames. For example, if there is no motion between two frames then very little information needs to be transmitted each successive frames. In areas of videos with more motion more pixels change from one frame to the next. When more pixels change, the algorithm must send more information or bits to keep up with the larger number of pixels that are changing.

Usually in video coding, the frames are processed in groups. One frame (usually the first) is encoded without motion compensation just as a normal image. This frame is called I-frame or I-picture. The other frames are called P-frames or P-pictures and are predicted from the I-frame or

P-frame that comes (temporally) immediately before it. The prediction schemes are, for example, described as IPPPP, meaning that a group consists of one I-frame followed by four P-frames.

Frames can also be predicted from future frames. In that case a future frame needs to be encoded before the predicted frame and so the encoding order does not necessarily match the real frame order. Such frames are usually predicted from two directions, i.e., from I or P-frames that immediately precede or follow the predicted frame. These bi-directionally predicted frames are called B-frames. A coding scheme could, for example, be IBBPBBP.

2.1 Motion Compensation

In video compression, motion compensation is a technique for describing a picture in terms of translated copies of macroblocks. This approach removes the redundancy between successive frames and thus improves compression efficiency.

A simple macroblock replacement scheme works only in the areas of the picture where a macroblock in the present frame and past frame are similar. But, most often the macroblock in the new frame and previous frame is not similar. However, the information contained in the new frame's macroblock is available in the previous frame, but it has moved. For example consider a moving car across the screen. Even though the object has moved from one frame to the other, much of the data required for the latest frame is still available in the in the previous frame, although not in the same place. Motion compensation tries to achieve greater compression by reusing portions from the previous frame to construct the new frame, even if they have moved.

In video coding the reference frame is subtracted from a given frame. The difference is then called the residual and usually it contains less energy or information than the original frame. The residual can be encoded at a lower bit rate with the same quality. The decoder can reconstruct the original frame by adding the reference frame again. The motion compensation tries to make the above process more efficient.

As we have discussed earlier a simple subtraction can only work in areas where there is no motion in the picture. Therefore, a more efficient approach would be to approximate the motion of the whole scene and objects of a video sequence. Then the subtraction process is carried with the translated reference block. This process can be simply defined with the following equation.

$$e_{x,y} = Current\_MB_{(x,y)} - \mathrm{Re}f\_MB_{(x',y')}$$

Where,

e is the error in the coordinate x,y

and x',y' is the translated coordinate of the reference.

This process gives a better residual but of course the bits occupied by the motion model should not be very high.

2.1.1 Global Motion Compensation

Global motion compensation tries to model the global motion of the video. It usually reflects camera motions such as dolly (forward, backward), track (left, right), boom (up, down), pan (left right), tilt (up, down) and roll (along the view axis). It works best for still scenes without moving objects. There are several advantages of global motion compensation:

- It models precisely the major part of motion usually found in video sequences with just few parameters. The bits consumed to describe the motion are negligible.
- It does not partition the frames thus avoiding artifacts. This also does not introduce any discontinuities in the time axis.

The major drawback of this scheme is that, it can not model the moving objects. Also, the use of global motion compensation along with block based motion compensation offers little advantage.

2.1.2 Block Motion Compensation

In block motion compensation, the frames are partitioned into smaller rectangular blocks known as macroblocks. Each macroblock is predicted from a macroblock of equal size in the reference frame. The blocks are not transformed in any other way apart from being shifted to the position of the predicted block. This shift is represented by a motion vector.

Since the macroblock size is usually small, in most of the video pictures a moving object is covered by multiple macroblocks. This fact is exploited while encoding the motion vectors. While encoding usually only the difference between the current and previous motion is used. This reduces the number of bits required to encode the motion vectors significantly.

It is possible to shift a macroblock by a non integer number of pixels (also known as pels in short form). This is usually called a sub-pixel precision. The in between pixels are generated by interpolation. Most commonly half-pixel or quarter-pixel precision is used.

The major disadvantage of the block motion compensation is that it introduces discontinuities at the block borders. Usually this calls for post processing or smoothing operations to be done either at the encoder or decoder.

Block motion compensation divides the current frame into non overlapping blocks and the motion compensation vectors tell from where these block come.

2.1.3 Variable Block Motion Compensation

Variable block motion compensation is essentially a block motion compensation technique in which the encoder can select the size of macroblocks dynamically. In terms video coding, the larger blocks can reduce the number of bits needed to represent the motion vector while smaller blocks can reduce the prediction residual. So, these encoders allow the optimal selection of macroblock sizes depending on the scene. The recent encoders such as H.264 etc. give the ability to dynamically select the macroblock size.

2.1.4 Overlapped Block Motion Compensation

As we have discussed earlier, the block based approach of motion compensation gives rise to several artifacts. Overlapped block motion compensation is a good solution to these problems. This scheme allows for a more prediction accuracy associated with no blocking artifacts.

In this method, the macroblocks are typically doubled in each dimension and overlap quadrant-wise with all neighboring blocks. Thus, each pixel belongs to 4 blocks. This means that each pixel has 4 predictors and these are summed up to a weighted mean. For this purpose, the blocks are associated with a window function. The major drawback of this method is computational complexity. The H.264 Annex F uses a modified version of this approach.

3. MOTION ESTIMATION

Motion estimation is the process of finding optimal (or near optimal) motion vectors. In other words, this means to find the macroblock in the reference which can be used for predicting the current macroblock. The optimal macroblock in the reference frame is selected based on some sort of minimization of a cost function. The most commonly used cost function is mean squared error (MSE) or sum of absolute difference (SAD) between the predicted and actual pixel values over all pixels of the motion -compensated region or the macroblock.

The motion estimation algorithm basically finds the block prediction error for each motion vector within a certain search range and picks the best compromise between the amount of error and number of bits needed for motion vector data. The motion estimation algorithm which exhaustively tests all possible combinations of motion representation is known as a full search. Though the full search gives the optimal motion vector it may not be a practical scheme for a real time encoder. Another approach in which the algorithm optimizes on the number of searches to find a possible candidate is known as fast search. The fast search algorithms are usually inferior to the full search with respect to the rate-distortion. We explore four most motion estimation algorithms below. These are Full Search, Diamond Search, and Predictor Search.

3.1 Full Search Motion Estimation

In Full Search (FS) ME, all possible overlapping locations around the collocated macroblock in the reference frame [Fig 2] are evaluated. The position which gives the minimum cost function (e.g. SAD) is selected as the winner predictor. In the exhaustive full search, the search area spans whole of the reference frame. But usually the search region is restricted into a limited region in the reference region as shown in the Fig 2. In general Full Search gives the most optimal predictor especially if one uses exhaustive search or the search region is sufficiently big.

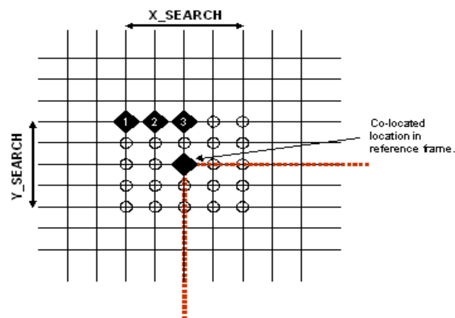

Fig 2 Full Search Motion estimation

3.2 Diamond Search Motion Estimation

Diamond Search algorithm [Fig 3] was proposed by Shan Shu and Kai-Kuang Ma [1]. This algorithm is designed based on a statistical study of motion vector fields. The diamond pattern is derived from the probability distribution function as those locations corresponding to the highest probability of finding the matching block in the reference frame.

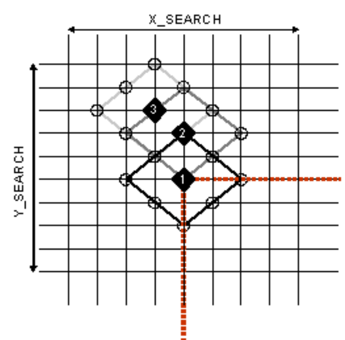

Fig 3 Diamond Search Motion estimation

The algorithm starts in the co-located macroblock in the reference frame using is as the center of the Diamond as shown in the fig. 4. It calculates total 9 SADs for each of the Diamond. Once the minimum of the SADs are found the centre of the Diamond is displaced to the minima and a new search is started. The search stops once the minima is the center of the Diamond.

The Diamond search takes fewer SADs when compared to a Full Search. But, as the Diamond Search is sparse search method it can totally skip an optimal point which may fall inside the diamond.

3.3 Predictor Search Motion Estimation

Predictor Search or Predictor based search is a general class of fast motion estimation algorithms which is based on the principle that, these algorithms uses past predictors to estimate the initial search region. These algorithms use neighboring blocks and blocks on adjacent frames with already computed motion vectors to estimate the initial position of the search. Once the initial position is found, it uses a regular search method like Diamond or Full search to further refine the candidate. A typical predictor based search is depicted in Fig 4. It uses a Diamond search for the final refinement.

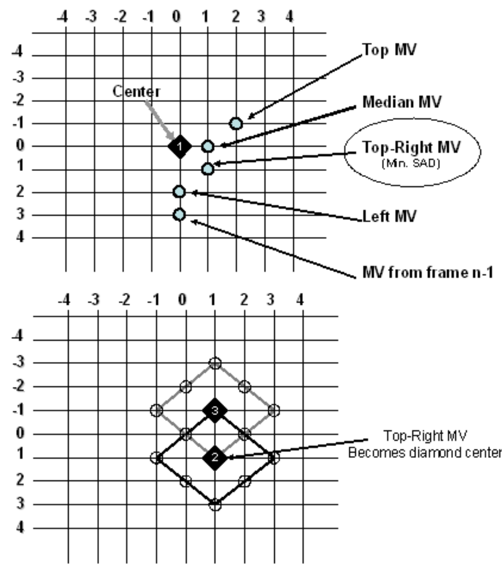

Fig 4 Predictor Search Motion estimation

4. REAL TIME IMPLEMENTATION CONSIDERATIONS

In the above sections we talked about the various algorithms for Motion Estimation. In this section we will discuss the various factors one needs to take care while using these algorithms for real time video encoding.

The motion estimation algorithms can be classified into two dimensions of complexity. The first dimension is the computational complexity and the second being the memory access complexity. We will talk about both of these factors in the light of the algorithms we discussed in the previous section.

4.1 Computational Complexity

The major computational complexity of a motion estimation algorithm arises due to the complexity of computing the cost function. Usually there are two kind of cost functions used.

The first one is Sum of Squared error and the second is the Sum of Absolute difference. Both the error measures are defined below.

$$SSE = \sum_{0}^{n*m-1} (X_{(n,m)} - X'_{n',m'})^2$$

$$SAD = \sum_{0}^{n*m-1} | (X_{(n,m)} - X'_{n',m'}) |$$

Where,

n is the number of rows,

m is the number columns,

X is the pixel in the position (n,m) in the current picture and

X' is the pixel in the reference frame.

In the research community there is a lot of debate on which one of above is a better error function. But as we can see SAD is simpler to the SSE as the requirement of square is not there. So, in case the hardware does not support the direct implement of SSE the SAD is usually the preferred cost function.

Irrespective of the cost function, the most important criterion that determines the overall complexity of the algorithm is: at how many points the cost function is evaluated.

In case of the Full search algorithm we have seen that it does evaluate the SAD at every point of the grid. Therefore the computational complexity increases as square of search range. In most realistic scenes the motion search range should be around +/- 64. For this search range the full search algorithm will require 128*128 SAD computations for each macroblock. This number is very high for most of processor available for real time applications.

Because of the higher computational complexity, in the real time applications full search algorithms are not used or it is only used in a final search stage with a very small search range. In most real time applications the ME is usually designed as two stages.

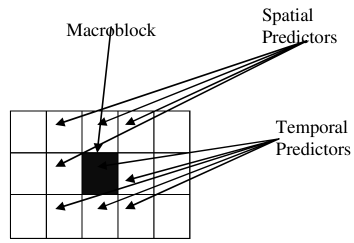

Fig 5 Predictor Selection for ME

The first stage is used to narrow down the search into a small region in the reference frame. In the first stage one can use a Diamond search. As we have seen in a Diamond search only 9 SADs are required to be calculated. This reduces the computational complexity significantly. The designer has the freedom of choosing the distance between two SADs in a Diamond. Usually it is compromise between complexity and search area coverage. The Diamond search can also be supported by predictors. The predictors are usually a very good indictor of overall motion in natural sequence. In most of the applications one chooses equal number of temporal and spatial predictors. For example if one decides to use all neighboring predictors, then he should use 4 from the last frame (temporal) and four from the current frame (spatial). This is depicted in the Fig 5.

The second stage usually consists of a very small full search of the winner from the first stage. This stage also does additional calculations of the ½ pel and ¼ pel refinement

4.2 Memory Access Complexity

For motion estimation one needs two data points available for the calculations. One is the macroblock for which the ME is being calculated. The other is the reference region. The reference region is the bigger chunk of data required. In general, most of the embedded processors do not have enough internal memory to keep the whole of the current frame and the reference frame. For this region the current as well as the reference regions are kept in the external memory.

Due to the above said reason, the data has to be accessed either via cache or dynamically brought to the on chip internal memory. In most of the embedded real time application the cache approach is not used due to the performance considerations. The most preferred approach for real time design is to use the internal memory for computations. In order to do that the data has to be brought to the internal memory and usually a DMA is used for this.

Since the internal memory is limited this puts a limitation on the search area one can use. The most preferred approach getting the data to the internal memory is to use a sliding window. This approach minimizes the data transfer while allowing a very good search range. Usually most real time design uses this approach. This is shown in the fig 6.

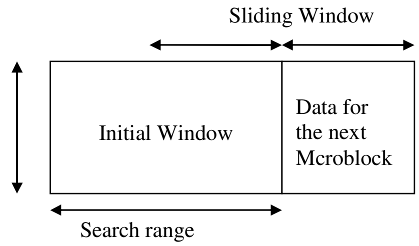

Fig 6 Sliding Window Reference Region ME

The sliding size of the window is determined by the DMA bandwidth of the processor.

5 SUMMARY

In this paper we talk about motion compensation and estimation. We also try to cover the limitations care about for a real time implementation of a motion estimation algorithm.

6 REFERENCES

[1] Zhu, S. et al. A New Diamond Search algorithm for Fast Block Matching Motion Estimation. ICICS, IEEE, Sept. 1997.

[2] Tourapis, A. et al. Predictive Motion Vector Field Adpative Search Technique (PMVFAST). Dept. of Electrical Engineering, The Hong Kong University of Science and Technology.

ABOUT THE AUTHOR

Pankaj received his B.E. from Sarder Vallabhbhai National Institute of Technology, Surat and M.Tech, from Indian Institute of Science, Bangalore. He is with the Multimedia Codec group in Texas Instruments Bangalore and is a Member of Technical Stuff. His current interests are Video quality improvement for real time application, speech processing and noise suppression.

UPCOMING EVENTS

- TI Developer Conference 2007 (India) is just around the corner. It brings TI's technology experts and TI's ecosystem firms together for two days to allow you to experience and evaluate a wide range of developments across DSP, microcontroller and analog technologies, enabling the rapid transformation of ideas and innovations into ready-to-use systems and solutions. For more information visit www.ti.com/tidcindia07

- The International Conference on IP Multimedia Subsystems Architecture and Applications-2007 (IMSAA-2007) will be held on 6th-8th December 2007 at Bangalore, India. This event is brought to you by IEEE Bangalore-Section and International Institute of Information Technology - Bangalore (iiit-b). For more details visit http://www.iiitb.ac.in/imsaa2007/index.html

- The Indian Institute of Technology Madras invites you to the Winter School in Speech and Audio Processing 2008 (WiSSAP-2008) to be held at ICSR Auditorium, I.I.T. Madras, Chennai, during 02-05 January, 2008. For further details: http://www.cse.iitm.ac.in/~wissap08

RESOURCE CORNER

- If you are into statistical modelling or otherwise use random number generators in your simulations or products, you know the extreme difficulty in producing truly random numbers. We often satisfy ourselves with quasi-random numbers, thereby compromising the quality of our endeavours. Well, you don't have to do that anymore. There is now a service that will generate truly random numbers for you!

  http://random.irb.hr/

- The Embedded Reading Room brings you selections from recently published books on engineering and programming .The selections are updated weekly. This remarkable service is absolutely free.

  http://www.embedded.com/197001133?_requestid=653444

- Those who work in the field of packet voice may have already noticed that conventional approaches to QoS are not always applicable in the VoIP space. Here is an article that talks about this aspect

  http://www.commsdesign.com/showArticle.jhtml;jsessionid=NVQGMJQINHZXMQSNDLRCKH0CJUNN2JVN?articleID=201001377

- Read a classic Machine Vision paper by Dr.Gerard Medioni at

  http://www.ddj.com/embedded/199201745?cid=RSSfeed_DDJ_EmbeddedSystems

  This has been hailed as the most influential machine vision paper of the last decade.

- Dolby® Volume™ is a new technology that controls the volume of audio across applications and media types. The technology is based on psycho-acoustic models and is designed to avoid volume discrepancies experienced by the user while changing channels, discs, media format etc. Read about it at

  http://www.dolby.com/assets/pdf/professional/pro_audio_engineering/Dolby_Digital_and_Dolby_Volume_Loudness_Solution.pdf

- This free software enables you to analyze the audio response of an audio component or system through a generic sound card (or through an external audio I/O device).

  http://www.audioroot.net/index.html?analysis/mataa.html

GETTING CONNECTED

Mailing list for IEEE SP Bangalore chapter

To facilitate better information spread across the wide spectrum of members and volunteers of the SP Bangalore chapter, a LISTSERV list is available.

- To email all of the list's subscribers (please use this responsibly), send your mail to IEEESP-BLR@LISTSERV.IEEE.ORG

- Creating a new subscription is easy. If you want to subscribe a member to the list, send a mail from your email to LISTSERV@LISTSERV.IEEE.ORG and type "subscribe ieeesp-blr" without quotes in the body of the message. Leave the subject line blank. More information on using LISTSERV is available at http://listserv.ieee.org/

Links

This link contains information related to IEEE SP conferences
http://www.ieee.org/organizations/society/sp/SPSConf.html

IEEE SP Bangalore Chapter Homepage
http://ewh.ieee.org/r10/bangalore/sps/

CHAPTER INFORMATION

Current Chair
Ajit Rao, Texas Instruments

Current Secretary and Treasurer
Prabindh Sundareson, Texas Instruments, Bangalore

Membership Development
Amitav Das, Microsoft Research –India, Bangalore

Conferences Co-ordinator
Jwalant Desai, Wipro Technologies, Bangalore

EDITORIAL BOARD

Radha Meka, Wipro Technologies, Bangalore
(radha.meka@WIPRO.COM)

Shankar Paramasivam, Wipro Technologies, Bangalore
(shankar.paramasivam@WIPRO.COM)

Karthikeyan R, Teclever Solutions Pvt Ltd
(karthik@teclever.com)

Polaka Ramana Reddy, Bangalore
(ramana_541@yahoo.com)

Prabindh Sundareson, Texas Instruments, Bangalore
(Prabindh@yahoo.com)

Chetan Vinchhi, Texas Instruments, Bangalore
(chetanv@ti.com)