Low Complex Block Level Correlation and Registration for Video Frame Interpolation

Brahmadesam T Madav¹; S.A.K. Jilani²; S. Aruna Mastani³
¹Jawaharlal Technological University Anantapur (JNTUA), India.
²Madanapalle Institute of Technology and Sciences, Madanapalle, India.
³Jawaharlal Technological University Anantapur (JNTUA), India.

Abstract
In this paper, we propose a low complex block level correlation and registration for Video Frame Interpolation (VFI) algorithm. The proposed method finds the correlation between the corresponding non-overlapping blocks. The correlation coefficients indicate the amount of similarity between the blocks. Using the median of correlation coefficients as threshold, the blocks are blended using image registration process. Compared to existing block based Motion Compensated Frame Interpolation (MCFI) methods, proposed algorithm is a low complex algorithm. In the proposed, there is no need of Motion Estimation (ME), smoothing of unreliable Motion Vectors (MVs) and Motion Prediction. Extensive experiments are conducted on various benchmark video sequences. Subjective assessment shows that the artefacts like blocky artefacts, holes, blurry artefacts, and ghost effects in MCFI are overcome by this algorithm, as there is no ME, preserving the true motion of the object to a greater extent is possible. The metrics PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity Index Metric) provide the objective assessment.

Key-words: PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Index Metric), Motion Compensated Frame Interpolation (MCFI).

1. Introduction

Over the recent decades usage of Video technology has grown enormously. To transmit video frames over the band limited channels, every alternate frame is skipped, remaining frames are encoded, and then transmitted as per video coding standards viz., H.264/AVC and H.265/HEVC. One of the steps in encoding process employs skipping of the frames which are then reconstructed at the decoder on the receiving side. The process of reconstruction at the receiver is Frame Interpolation.
(FI). The FI technique uses the previous and the next, temporally related frames. The previous and next frames are partitioned into a group of pixels, called Macro Blocks (MB). The frame to be interpolated is reconstructed using the blocks from these previous and next frames. The motion of the objects between either block relates MBs to each other in the frame. The Block Matching Algorithms (BMAs) estimates the relationship in best matching process. The motion is represented by the Motion Vector (MV) for a particular block. Combination of Motion Vectors (MVs) of all blocks in a frame is Motion Estimation (ME). The frame is interpolated using ME. Popularly there are many techniques of FI that use ME. These are known as Motion Compensated Frame Interpolation (MCFI) techniques. Non-MCFI techniques are available in literature. The functions of decoder are (1) validating the received MVs, (2) identifying the correct MVs and (3) correcting the errored MVs normally known as smoothing of MVs. All the MCFI techniques work on improving the validity and correction of received MVs. Artifacts like blurring, ghost effects, holes, incorrect occlusions occur if there is an error in MVs. The artifacts are minimized by smoothing MVs.

Besides BMA, ME at the decoder is proposed to estimate MVs for smooth Motion Vector Filed (MVF). The Motion Vector Processing (MVP) techniques remove the outliers by refining MVs. The MVP results better in smaller regions where motion is smooth and regular. In BMA with MCFI, a single pixel can pass through multiple trajectories or none, resulting overlapped regions and holes in the interpolated frames, respectively. Many researchers worked at pixel level transformations for an interpolated frame by super imposing or blending of images. Optical flow method, one of the methods for identifying the flow field in images has also been used and proven better than many MCFI methods. This technique is highly computational complex.

Motivated to reduce computational complexity and utilize block level correlation, we propose this algorithm. Our algorithm cross correlates corresponding blocks and blend the selected blocks through registration. In Section-2 we discuss in detail about our proposed method. In Section-3, we discuss about the Subjective analysis and Objective analysis.

2. Proposed Algorithm

In our method, first we adopted block level correlation, second a threshold from correlation coefficients is selected and third based on correlation threshold blending of corresponding blocks is performed by registration.
Block based methods can operate on large region of the frame with low complexity. The correlation coefficient represents the similarity factor between the blocks. If the similarity between the blocks is high the correlation coefficient is maximum and minimum when the blocks are dissimilar. We intend to use the low complexity property of block based operation along with correlation to reduce computational complexity. The blocks are made such that, no overlapping blocks occur in a frame. Overlapping blocks results in incorrect correlation and missing of regions when frame is interpolated. The non-overlapping process avoids the reuse of the regions thereby decreasing complexity.

The median of correlation coefficients is selected threshold. The blocks are grouped based on correlation value. The blocks with correlation coefficients greater than threshold are grouped as GTT, while the block’s coefficient less than threshold as LTT. The blocks under GTT category in the interpolated frame are taken from the next frame. The blocks of interpolated frame under LTT category are obtained by blending corresponding blocks of previous and next frame. The block of
GTT and LTT group of, to be interpolated frame, are concatenated for complete interpolated frame. The interpolated frame is compared with original frame in terms of PSNR and SSIM metric. Also computed computational complexity.

3. Results and Discussion

We experimented our algorithm on CIF video sequences viz., Bus, Foreman, Mobile and Soccer. Using our proposed algorithm, we interpolated even frames using odd frames. The PSNR and SSIM metrics were shown graphically for a video of 1 second @ 30 fps. From a video of 1 second duration, we interpolated 15 frames. Each frame was interpolated using block sizes of 16, 32, 64 and 128 pixels for CIF format. The graphs represent the PSNR and SSIM for each interpolated frame at various block sizes.

3.1. Subjective Analysis

In this paper, we compared our proposed method with previous Block Based methods. The abbreviation is given in brackets for simplicity. The methods were Bi-directional Interpolation using Vector Median Filtering (Bi-MCI + VMF) [2][6], Correlation Based MV Processing with Adaptive Interpolation (CMVP + AI)[5]. Multilevel Video Frame Interpolation (MVFI)[1] and Optical Flow Interpolation (OFI)[9] for Foreman and Bus video sequences. For Mobile and Soccer video sequences, we compared our results with Multi-frame Based Occlusion Handling (MBOH)[10], Frame Interpolation using Temporal Information (TI)[7], Novel True Motion Estimation (NTME) [3], and Region based MCFI (RB) [8] methods.

Figure 2 - Interpolated Frames of Foreman Video Sequence

Figure 2(a)  Figure 2(b)  Figure 2(c)  Figure 2(d)
In the figure (2) and (3) the 14th frame of Foreman sequence in our interpolation is compared with Bi-MCI+VMF and CMVP+AI. From the figure 2(a) and 2(b) it can be observed that in blocky artifacts at nose predominantly. These artifacts are due to incorrect MVs. The CMVP + AI, produce better results than the other, but still blocky artefacts are still visible at nose and right eye. From figure 2(c) and 2(d) ghost artefacts can be observed at top of the hat, top and bottom of the ear, nose, right eye, mouth, eyebrow, and collar produced by MVFI[1] algorithm. The interpolation created ghost artifacts due to over smoothing of VMF. In, 2(c) blurriness can be seen at teeth, but reduced distortion at the cost of increased complexity by OFI technique[9]. In figure (3) such no artifacts are visible, and the original structure had been retained by our proposed method.
In figure 4(a), interpolation results of 22nd frame of Bus video sequence by Bi-MCI + VMF [2] [6] technique is shown. This method produces distortions at front top of the bus, bus stop post, bottom of the lamp post and railings at the left most part as shown. In 4(b), CMVP +AI method of FI blurriness is more prominent at the left portion of the railing. In 4(c) MVFI method, at the right, trees at the top are less blurred, and in 4(d), OFI method blurriness can be noticed at persons and statue before bus, the lamp post has blocky artefacts. In figure.(5), our proposed method avoided these artefacts.

In figure 6(a), artefacts can be observed at numbers in bottom line, 58th frame in Mobile video sequence by MBOH method. In figure 6(b) and in 6(c) TI and NTME methods of frame interpolation produced blurriness around the ball and small objects. RB method, as shown in figure 6(d) produce artifacts due to region based considerations. In our method, as shown in figure.7, these artifacts were avoided.

Figure 6 - Interpolated Frames of Mobile Video Sequence

Figure 6 (a)  Figure 6(b)  Figure 6(c)  Figure 6(d)

Figure 7 - Interpolated 58th Frame of MOBILE Video by Proposed Method

It can be observed from figure 8(a) and 8(b), MBOH and RB methods resulted artifacts that were predominant at the foot of the right side player and ball. From figure 8(c), TI method blocky artifacts were visible at the head of left side player. From figure 8(d), NTME method produced
artifacts in the background. Our proposed algorithm resulted in figure 9, in which the artifacts were minimized for 159th frame of Soccer video sequence.

Figure 8 - Interpolated Frames of Soccer Video Sequence

Figure 8 (a) Figure 8(b) Figure 8(c) Figure 8(d)

Figure 9 - Interpolated Frame Soccer 159th Frame by Proposed Method

3.2. Objective Analysis

From the Foreman CIF, as in figure 10(a), frame 28th of the interpolated frame had the highest PSNR with block size of 16 pixels. The other blocks produced PSNR with a difference of 0.002 by 32 pixel, 0.123 by 64 pixel, and a 128 pixel block by 0.77. From this difference, we could conclude that at 16 and 32 pixel block sizes, our algorithm produced almost same result. The 64 and 128 block sizes produced a bit different result. From figure 10(b) Bus CIF, 2nd interpolated frame had the highest PSNR values at 128 pixel block size, showing that the motion of the object was traced smoothly at 128 block sizes. The difference with other blocks is 0.011,0.012 and 0.009 of 16, 32 and 64 block sizes, respectively.
Table 1 - PSNR of Foreman and Bus Video Sequences

| FRAME | FOREMAN (Block Size) | BUS (Block Size) |
|-------|----------------------|------------------|
|       | 16 Pixel | 32 Pixel | 64 Pixel | 128 Pixel | 16 Pixel | 32 Pixel | 64 Pixel | 128 Pixel |
| 2     | 26.5666  | 26.5666  | 26.5256  | 26.5464  | 16.7786  | 16.7785  | 16.7788  | 16.7797  |
| 4     | 27.0852  | 27.0843  | 27.0183  | 27.0496  | 16.4565  | 16.4566  | 16.4553  | 16.4587  |
| 6     | 29.4978  | 29.4964  | 29.3633  | 29.4427  | 16.3145  | 16.3138  | 16.3150  | 16.3158  |
| 8     | 27.1395  | 27.1371  | 27.0940  | 27.1208  | 16.1697  | 16.1668  | 16.1636  | 16.1664  |
| 10    | 27.5380  | 27.5371  | 27.4893  | 27.5173  | 15.9017  | 15.9020  | 15.8958  | 15.8981  |
| 12    | 25.4335  | 25.4334  | 25.3993  | 25.4191  | 15.9166  | 15.9188  | 15.9132  | 15.9167  |
| 14    | 24.1519  | 24.1513  | 24.1338  | 24.1378  | 16.0502  | 16.0549  | 16.0449  | 16.0483  |
| 16    | 25.4776  | 25.4755  | 25.4529  | 25.4575  | 15.9517  | 15.9494  | 15.9465  | 15.9506  |
| 18    | 27.3022  | 27.2998  | 27.2628  | 27.2767  | 15.8755  | 15.8701  | 15.8711  | 15.8728  |
| 20    | 28.5750  | 28.5729  | 28.5144  | 28.5445  | 15.9496  | 15.9496  | 15.9481  | 15.9485  |
| 22    | 23.5928  | 23.5924  | 23.5878  | 23.5903  | 16.0479  | 16.0483  | 16.0469  | 16.0463  |
| 24    | 23.3414  | 23.3409  | 23.3421  | 23.3375  | 15.9694  | 15.9728  | 15.9692  | 15.9715  |
| 26    | 27.8475  | 27.8479  | 27.7918  | 27.8236  | 15.6980  | 15.6976  | 15.6955  | 15.6960  |
| 28    | 31.2549  | 31.2534  | 31.1317  | 31.1785  | 15.7004  | 15.7004  | 15.7028  | 15.7020  |
| 30    | 28.9599  | 28.9624  | 28.9068  | 28.9169  | 15.6883  | 15.6895  | 15.6897  | 15.6898  |

Figure 10 (a) and 10 (b) - PSNR of Interpolated Frames at various Block sizes of Foreman and Bus Video Sequences

Considering figure 11(a), Mobile CIF, 2nd interpolated frame had the highest PSNR with 16 pixel block size. The difference with other blocks is 0.005,0.15 and 0.024 with respect to 32, 64 and 128 blocks, respectively. For Soccer CIF, from figure 11(b), 8th interpolated frame had better PSNR,
for proposed method, with 16 and 32 pixel block sizes. While the difference was 0.1228 and 0.0142 with respect to 64 and 128 pixel blocks.

Table 2 - PSNR of Mobile and Soccer Video Sequences

| FRAME | MOBILE (Block Size) | SOCCER (Block Size) |
|-------|---------------------|---------------------|
|       | 16 Pixel | 32 Pixel | 64 Pixel | 128 Pixel | 16 Pixel | 32 Pixel | 64 Pixel | 128 Pixel |
| 2     | 19.9270  | 19.9226  | 19.9113  | 19.9027  | 23.1011  | 23.1002  | 23.0300  | 23.0867  |
| 4     | 19.8664  | 19.8596  | 19.8484  | 19.8415  | 23.5973  | 23.5974  | 23.4992  | 23.5845  |
| 6     | 19.6967  | 19.6900  | 19.6801  | 19.6748  | 23.8759  | 23.8758  | 23.7940  | 23.8631  |
| 8     | 19.6615  | 19.6581  | 19.6441  | 19.6370  | 24.2479  | 24.2479  | 24.1251  | 24.2337  |
| 10    | 19.3056  | 19.3039  | 19.2964  | 19.2949  | 24.0772  | 24.0773  | 23.9583  | 24.0710  |
| 12    | 19.4430  | 19.4249  | 19.4171  | 19.4093  | 23.7894  | 23.7886  | 23.6480  | 23.7716  |
| 14    | 19.4385  | 19.4276  | 19.4272  | 19.4180  | 23.3728  | 23.3727  | 23.2909  | 23.3590  |
| 16    | 19.6373  | 19.6328  | 19.6288  | 19.5973  | 23.4878  | 23.4878  | 23.3934  | 23.4769  |
| 18    | 19.7992  | 19.7885  | 19.7897  | 19.7772  | 23.8119  | 23.8109  | 23.7107  | 23.7999  |
| 20    | 19.7780  | 19.7771  | 19.7648  | 19.7481  | 23.5763  | 23.5761  | 23.4918  | 23.5626  |
| 22    | 19.3886  | 19.3737  | 19.3687  | 19.3535  | 23.1259  | 23.1260  | 23.0431  | 23.1167  |
| 24    | 18.7789  | 18.7624  | 18.7529  | 18.7372  | 23.0683  | 23.0680  | 22.9916  | 23.0583  |
| 26    | 18.4770  | 18.4708  | 18.4712  | 18.4611  | 22.2619  | 22.2609  | 22.2229  | 22.2503  |
| 28    | 18.8924  | 18.8847  | 18.8657  | 18.8580  | 21.3242  | 21.3241  | 21.2972  | 21.3191  |
| 30    | 19.4325  | 19.4194  | 19.4127  | 19.3940  | 21.2291  | 21.2291  | 21.2165  | 21.2226  |

Figure 11 (a) and 11 (b) -PSNR of Interpolated Frames at various Block sizes of Mobile and Soccer Video Sequences
The SSIM comparison of Foreman is shown in figure 12(a) 6th interpolated frame of the proposed method produced best SSIM at 16 and 32 pixel block sizes. The difference in SSIM was 0.030 and 0.012 for 64 and 128 block sizes. In the Bus SSIM comparison in figure, 12(b) the 22nd interpolated frame had best SSIM at 16 and 32 pixel block size. The difference with other block sizes 64 and 128 was 0.006. As shown in figure.

| FRAME | FOREMAN (Block Size) | BUS (Block Size) |
|-------|---------------------|-----------------|
|       | 16 Pixel | 32 Pixel | 64 Pixel | 128 Pixel | 16 Pixel | 32 Pixel | 64 Pixel | 128 Pixel |
| 2     | 0.9490   | 0.9490   | 0.9473   | 0.9481    | 0.4230   | 0.4224   | 0.4224   | 0.4224    |
| 4     | 0.9505   | 0.9505   | 0.9479   | 0.9492    | 0.4213   | 0.4222   | 0.4220   | 0.4218    |
| 6     | 0.9711   | 0.9711   | 0.9681   | 0.9699    | 0.4314   | 0.4315   | 0.4312   | 0.4305    |
| 8     | 0.9378   | 0.9378   | 0.9358   | 0.9369    | 0.4314   | 0.4319   | 0.4310   | 0.4308    |
| 10    | 0.9402   | 0.9402   | 0.9382   | 0.9394    | 0.4415   | 0.4422   | 0.4415   | 0.4417    |
| 12    | 0.9228   | 0.9228   | 0.9208   | 0.9218    | 0.4313   | 0.4315   | 0.4310   | 0.4317    |
| 14    | 0.9209   | 0.9208   | 0.9191   | 0.9198    | 0.4492   | 0.4496   | 0.4492   | 0.4493    |
| 16    | 0.9196   | 0.9195   | 0.9179   | 0.9183    | 0.4441   | 0.4435   | 0.4436   | 0.4434    |
| 18    | 0.9364   | 0.9364   | 0.9346   | 0.9352    | 0.4379   | 0.4374   | 0.4371   | 0.4373    |
| 20    | 0.9480   | 0.9480   | 0.9461   | 0.9470    | 0.4481   | 0.4483   | 0.4477   | 0.4476    |
| 22    | 0.8719   | 0.8719   | 0.8705   | 0.8710    | 0.4556   | 0.4556   | 0.4550   | 0.4550    |
| 24    | 0.8634   | 0.8634   | 0.8620   | 0.8623    | 0.4515   | 0.4525   | 0.4514   | 0.4517    |
| 26    | 0.9399   | 0.9399   | 0.9380   | 0.9389    | 0.4531   | 0.4529   | 0.4526   | 0.4527    |
| 28    | 0.9691   | 0.9691   | 0.9667   | 0.9680    | 0.4530   | 0.4530   | 0.4530   | 0.4528    |
| 30    | 0.9539   | 0.9539   | 0.9515   | 0.9527    | 0.4468   | 0.4468   | 0.4463   | 0.4460    |

Figure 12 (a) and 12 (b) - SSIM of Interpolated Frames at various Block Sizes of Foreman and Bus Video Sequences
In Mobile video sequence, as shown in figure 13(a), 20\textsuperscript{th} interpolated frame had the highest SSIM at 32 pixel block size. The difference is 0.006 at 16 and 64, 0.008 at 128 block sizes. The SSIM in Soccer interpolated frames, as in figure 13(b), frame 8\textsuperscript{th} interpolated frame at 16 and 32 pixel block size, had the highest, with 0.030 and 0.004 at 64 and 128 block differences.

| FRAME | MOBILE (Block Size) | SOCCER (Block Size) |
|-------|---------------------|---------------------|
|       | 16 Pixel  | 32 Pixel  | 64 Pixel  | 128 Pixel | 16 Pixel | 32 Pixel | 64 Pixel | 128 Pixel |
| 2     | 0.8259    | 0.8251    | 0.8232    | 0.8235    | 0.9214   | 0.9214   | 0.9195   | 0.9208    |
| 4     | 0.8255    | 0.8245    | 0.8227    | 0.8231    | 0.9341   | 0.9341   | 0.9313   | 0.9336    |
| 6     | 0.8255    | 0.8249    | 0.8238    | 0.8244    | 0.9355   | 0.9355   | 0.9334   | 0.9350    |
| 8     | 0.8301    | 0.8292    | 0.8274    | 0.8274    | 0.9438   | 0.9438   | 0.9407   | 0.9434    |
| 10    | 0.8115    | 0.8113    | 0.8102    | 0.8103    | 0.9427   | 0.9427   | 0.9397   | 0.9425    |
| 12    | 0.8095    | 0.8096    | 0.8074    | 0.8074    | 0.9359   | 0.9359   | 0.9326   | 0.9354    |
| 14    | 0.8169    | 0.8164    | 0.8143    | 0.8140    | 0.9273   | 0.9273   | 0.9250   | 0.9269    |
| 16    | 0.8270    | 0.8265    | 0.8245    | 0.8233    | 0.9261   | 0.9261   | 0.9236   | 0.9259    |
| 18    | 0.8364    | 0.8354    | 0.8334    | 0.8336    | 0.9318   | 0.9318   | 0.9293   | 0.9316    |
| 20    | 0.8407    | 0.8413    | 0.8401    | 0.8399    | 0.9255   | 0.9255   | 0.9231   | 0.9254    |
| 22    | 0.8207    | 0.8192    | 0.8191    | 0.8186    | 0.9147   | 0.9147   | 0.9123   | 0.9145    |
| 24    | 0.7898    | 0.7889    | 0.7863    | 0.7867    | 0.9146   | 0.9146   | 0.9123   | 0.9144    |
| 26    | 0.7799    | 0.7793    | 0.7780    | 0.7777    | 0.8915   | 0.8915   | 0.8901   | 0.8914    |
| 28    | 0.7976    | 0.7969    | 0.7952    | 0.7946    | 0.8601   | 0.8601   | 0.8589   | 0.8601    |
| 30    | 0.8237    | 0.8229    | 0.8228    | 0.8218    | 0.8643   | 0.8643   | 0.8636   | 0.8641    |
With minimum computational complexity, our proposed algorithm could produce better interpolated frames on par with existing FI techniques, with minimum artifacts. From fig.3, it can be observed that the PSNR of interpolated Foreman frame is 23.8073 and SSIM is 0.91046. SSIM value being close to Human Visual System (HVS) we could achieve best result when the motion in the frame is medium and low. In a fast and multiple objects motion our algorithm produced nominal results. Example is Bus frame in fig.5, with PSNR of 16.047 and SSIM of 0.45558. For medium motion frame like Mobile frame, PSNR is 20.7739 and SSIM is 0.87403 as shown in fig.7. Our algorithm performed best for Soccer video with PSNR and SSIM of 25.286 and 0.94172 respectively as in fig.9.
4. Conclusion

In this paper, we proposed a new VFI method based on Cross correlation at block level which reduces the number of block level operations. The blocks that are selected are blended using registration at block level. This reduces the computational complexity which is the prime objective of our method. We compared our proposed algorithm with existing FI methods. Our subjective comparison shows better results than that of existing. In this work, slow and medium motion objects are interpolated more accurately than fast moving objects. It is one of the future topics to develop a low complex block correlation based fast motion interpolation.

References

Zhefei Yu et al. “Multilevel Video Frame Interpolation: Exploiting the Interaction Among Different Levels”, *IEEE Transactions on Circuits and Systems for Video Technology*, 2013, 1235–1248.

S.H. Lee et al B.T. Choi. “New frame rate up-conversion using bi-directional motion estimation”, *IEEE Transactions on Consumer Electronics*, 2000, 603–609.

S. Dikbas and Y. Altunbasak. “Novel True Motion Estimation Algorithm and Its Application to Motion-Compensated Temporal Frame Interpolation”. *IEEE Transactions on Image Processing*, 2013, 2931–2945.

P. Heo G. Choi and H. Park. “Triple- Frame -Based Bi-directional Motion Estimation for Motion Compensated Frame Interpolation”, *IEEE Transactions on Circuits Systems for Video Technology*, 2019.

A.M. Huang and T. Nguyen. “Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation”, *IEEE Transaction on Image Processing*, 2008, 740–752.

P. Haavisto et al J. Astola. “Vector median filters”. *Conference Proc. IEEE*, 1990, 678–689.

D. Kim and H. Park. “An efficient motion-compensated frame interpolation method using temporal information for high-resolution videos”. *Journal of Display Technology*, 2015, 580–588.

H. Lim and H. Park. “A region-based motion-compensated frame interpolation method using a variance-distortion curve”. *IEEE Transactions on Circuits and Systems for Video Technology*, 2015, 518–524.

T. Pock et al M. Welberger. “Optical flow guided TV-L1 video interpolation and restoration”, *Proc. Energy Minimization Methods Computer Vision Pattern Recognition* (2011), 273–286.

N. Xu Q. Lu and X. Fang. “Motion-compensated frame interpolation with multi-frame based occlusion handling”. *Journal of Display Technology, 12.1* (Jan. 2016), 45–54.
Authors Profile

**B.T. Madav** is pursuing Full-time Research, Ph.D in JNTUA, Ananthapuramu at MITS, Madanapalle, India as Research Centre. He is graduated from IETE (117384), New Delhi, Post-graduated from JNTUK, Kakinada, India. He has industrial experience for 7 years and teaching experience of 15 years. His areas of interest include Image Processing, Video Processing, and Computer Vision. Mail id: madhavbt@gmail.com

**S.A.K Jilani** is currently Professor in Electronics and Communication Engineering (ECE), Madanapalle Institute of Technology and Science (MITS), Madanapalle, India. He received his Ph.D degree from Sri Krishnadevaraya University, Anantapur, India in 2002. His areas of interest are Image and Video Processing, Embedded Systems, Material Science. He is currently Principal Investigator of a DST project on Cognitive Science. He contributed many research papers to International, National Journals, Conferences and authored textbooks. Mail id: jilani_s_a_k@yahoo.com

**S. Aruna Mastani** is working as Assistant Professor in JNTU College of Engineering, Anantapuramu. She received B.E in Electronics and Communication Engineering from JNTU College of Engineering Anantapur and M.Tech in Digital Systems and Computer Electronics from JNTU College of Engineering, Anantapur. She received Ph.D degree in the area of Digital Image Processing from JNTUA, Anantapur. She worked as academic assistant in JNTUCE, Anantapur and worked as Assistant Professor and Associate Professor in Intell Engineering College Anantapur, She Published several papers in national and international journals and conferences in Image Processing, Signal Processing, and VLSI. Mail Id: aruna_mastani@yahoo.com