ROI BASED VIDEO COMPRESSION TAILORED FOR CONVERSATIONAL VIDEO COMMUNICATIONS

ECE 241 - PROJECT REPORT

Pradeep Kumar Govindaraju (pkg@umail.ucsb.edu)

Under the guidance of Dr. Jerry D. Gibson (gibson@ece.ucsb.edu)

Abstract:

In this report, Region of Interest (ROI) based resource allocation is done in Quantization stage of Video Compression to achieve psychovisual compression in video communication scenario. The Quantization depends heavily on the transform stage. The motion compensation stage depends heavily on both Transform and Quantization stage. So several experiments are done at different levels to see how ROI priority parameters affect each of these stages. For detecting ROI Viola Jones face detector is used. For the transform stage Integer transform and Discrete Cosine transform are used in parallel and their behavior along with ROI priority parameters is studied.

Introduction:

In conversational video communications human face regions are the ones that is mainly focused on. Thus if we consider face regions as regions of interest (ROI) and compress the video giving higher priority to ROI, we can get a perceptually good reconstruction of the video. This is highly necessary in the bandwidth constrained video coding. Also, in very low bit rate coding it is important to encode some image areas (e.g. faces) with higher fidelity than others (e.g. tree, leaves moving in the background). So the motion compensation should also be done based on ROI priority.

This problem is addressed in this paper by using region-of-interest (ROI) based quantization and motion estimation. The three major issues that will be addressed in this ROI based compression schemes are:

1) Detection and segmentation of ROI

2) ROI-based quantization scheme for basic coding units in a frame [ROI & non ROI priority]

3) Tradeoff between Quantization, Motion Compensation and ROI priority.

Problem Statement & Constraints:

The scenario under consideration is a real time video communication. So the video coding process must be in real time. This puts a constraint that detection and segmentation of ROI should be fast enough.

I. ROI detection algorithm – Viola Jones Face detection

1. Some algorithms utilize the visual features including motion to detect the ROI [1], [2]. But due to rate-distortion optimization, there is always a dilemma with applying motion to detect ROI because motion parameters can be obtained only after quantization and bit allocation.

2. Segmentation algorithms, some of which are almost real time can be applied to detect faces. Liu et al used skin color for ROI determination [3]. But this will not perform well with illumination changes.

3. Since face detection in still images is well studied, there were lot of learning/classification based methods widely considered for robust and real time face detection like Viola-Jones algorithm[4]. This makes Viola-Jones algorithm a better candidate for our scenario.

Representing ROI – Rectangular bounding box

The detected ROI can be represented by a bounding box or a shape mask depending on the algorithm being used for ROI detection. In case of segmentation based ROI detection, the ROI is of irregular shape and it might need a shape mask to represent [5]. In case of object detection based techniques, a rectangular bounding box represents the ROI. A rectangular window can be easily represented by storing the left top and right bottom pixel locations of the window whereas a non regular shape needs a shape mask, which needs several bits to store the complete shape mask. It is computationally heavy when it comes to 4x4 blocks for performing Integer transform, which cannot operate partially on non ROI and ROI pixels. An extra algorithm is required for checking the boundary of the ROI and non ROI regions and decision should be made whether the 4x4 block covering the boundary has to be considered as ROI block or not. Thus shape masks might be good for offline video compression, but in case of real time processing, rectangular window of ROI representation is better in terms of bit rate as well as computational complexity.

II. Forward Transformation and Quantization based on ROI.

The two most popular transform schemes are Discrete Cosine Transform and Integer transform. We will be looking at how both of these can be used for ROI scheme. It is to be noted that integer transform has the quantization stage clubbed with the transform stage. This makes integer transform to behave in a different manner compared to DCT where quantization (Quantization parameter) is independent of transform stage .

APPLYING ROI PARAMETERS INTO QUANTIZATION:

1. QUANTIZATION AFTER DCT:

As per MPEG standard, every quantized value of a macro block is

$$QF = (32 \times F)/(2 \times S \times W)$$

QF – Quantized coefficient
F – DCT coefficient
S – Quantization Scale
W – Quantization matrix element

So for giving higher priority to ROI and non ROI, introduce a ROI scale parameter, into the equation as follows.

$$QF = (32 \times F \times sROI)/(2 \times S \times W)$$

This sROI is kept high for ROI macroblocks and relatively low for non ROI macroblocks. The range of these parameters can be (0 – 2). This is obtained after trial and error with the above equation.

2. QUANTIZATION AFTER INTEGER TRANSFORM:

In Integer transform, we have control over only the Quantization Parameter (QP) value which defines all other parameters (like step size) as per standard. So we set lower QP value for ROI macroblocks and higher QP value for non-ROI macro block. This means, ROI priority is set using QP values for ROI and non ROI blocks.

The macro blocks considered here for the following Table1 are of size 4x4.

Note: Values highlighted in yellow means those values are fixed

TABLE 1 INTEGER TRANSFORM AND DISCRETE COSINE TRANSFORM-COMPARISON AFTER QUANTIZATION

	Input Image	ROI and non ROI Parameters	IT – Total (kilo bits)	DCT – Total (kilo bits)	IT-Comp. Ratio	DCT-Comp. Ratio	IT – ROI PSNR	DCT – ROI PSNR	IT-nonROI PSNR	DCT-nonROI PSNR	IT-Total PSNR	DCT-Total PSNR
1	Image1 – non ROI region is smooth	IT_ROI = 20 IT_nonROI = 35 DCT_ROI =1.45 DCT_nonROI = 0.13	154.32	108.13	0.158	0.110	43.59	45.04	35.67	35.64	36.90	36.95
2	Image1 – non ROI region is smooth	IT_ROI = 22 IT_nonROI = 35 DCT_ROI =0.915 DCT_nonROI = 0.13	146.95	92.37	0.150	0.094	42.03	42.02	35.67	35.62	36.78	36.74
3	Image1 – non ROI region is smooth	IT_ROI = 22 IT_nonROI = 51 DCT_ROI =0.915 DCT_nonROI = 0.23	124.94	98.70	0.127	0.101	42.03	42.02	24.71	38.34	26.18	39.14
4	Image2 – non ROI region is cluttered	IT_ROI = 20 IT_nonROI = 35 DCT_ROI =1.45 DCT_nonROI = 0.13	148.05	99.74	0.152	0.102	44.80	47.62	32.65	32.03	33.86	33.29
5	Image2 – non ROI region is cluttered	IT_ROI = 22 IT_nonROI = 35 DCT_ROI =0.915 DCT_nonROI = 0.13	145.64	89.83	0.15	0.092	44.10	44.13	32.66	32.02	33.88	33.24
6	Image2 – non ROI region is cluttered	IT_ROI = 25 IT_nonROI = 48 DCT_ROI =0.48 DCT_nonROI = 0.232	98.17	98.14	0.10	0.10	41.07	41.08	23.10	35.31	24.38	36.23

Observations:

1. For a fixed non ROI PSNR and Total PSNR, DCT is giving better ROI PSNR with better compression ratio as well.

2. For a fixed ROI, nonROI and Total PSNRs of DCT and IT, DCT is achieving slightly better compression.

3. For a fixed ROI PSNR, and trying to achieve similar Compression ratio, DCT has a better non ROI PSNR, which in turn makes the total PSNR better. It can also be noticed that Compression ratio of Integer transform is slightly higher than that of DCT. This is the minimum possible compression ratio that can be attained by Integer transform because the non ROI parameter is set to the max (51). This means ROI is compressed to the maximum allowed by h.264. Hence DCT outperforms Integer transform in case where ROI PSNR is fixed.

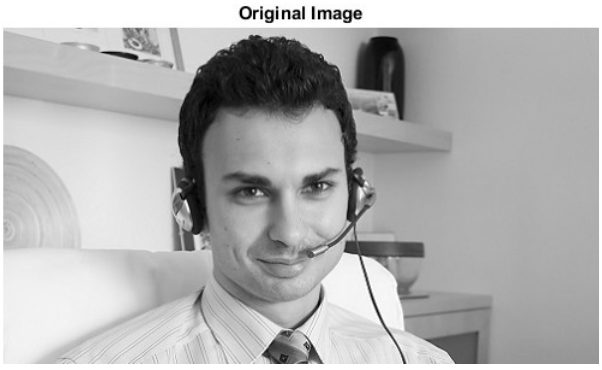
These three sets of experiments are now done on an image with a cluttered background. This means a lot of high frequency components to handle by the non ROI region.

Experiment 4,5,6 follows the same pattern as 1,2,3. DCT performs better than Integer transform. But in 4,5,6 Integer transform's performance is more close to that of DCT than the results in 1,2,3. This can be due to the fact that the background is more cluttered. In case of 1,2,3 the background is very smooth . This means it is more dominated by low frequency components and the flexibility in choosing quantization scales helped in fine tuning and get optimal performance.

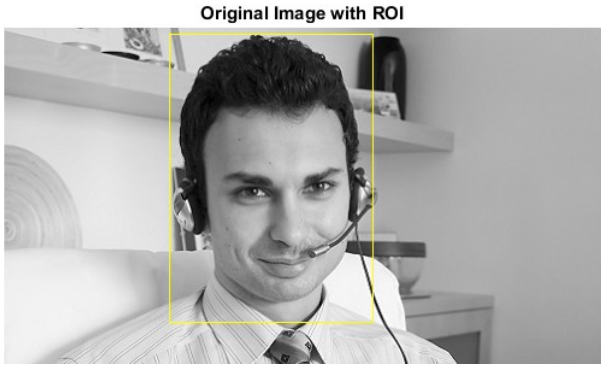
Also in experiment 6, where we fix compression ratios and ROI PSNR for DCT and Integer Transform, we can see that we are able to achieve equal compression ratios in DCT and Integer Transform. This was not even possible when we tried to do the same type of experiment with Image 1, which has smooth non ROI. It was in experiment 3.

Thus, the performance of integer transform comes very close to DCT when the high frequency component is high.

The following images are the originals and outputs of each of the experiments listed in Table 1.



Original for (1,2,3)



Original for (1,2,3) with ROI

DCT based Compression

(1)

Integer Tranform based Compression

(1)

DCT based Compression

(2)

Integer Tranform based Compression

(2)

DCT based Compression

(3)

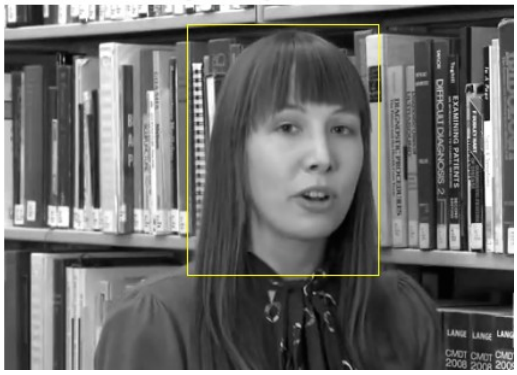
Integer Tranform based Compression

(3)

Original Image

Original Image-2

Original Image with ROI



Original Image-2 with ROI

DCT based Compression

(4)

Integer Tranform based Compression

(3)

DCT based Compression

(5)

Integer Tranform based Compression

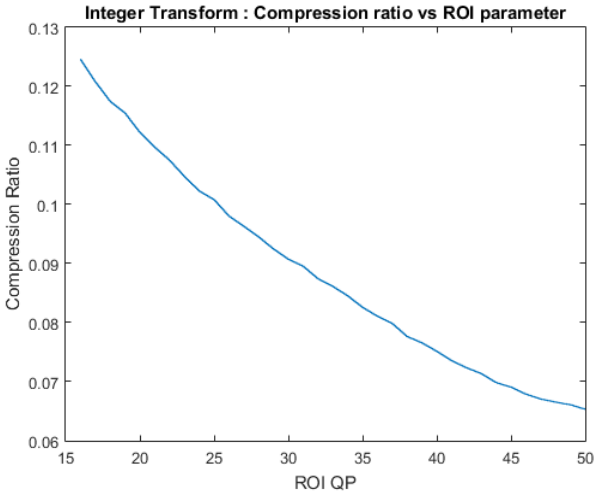
(5)

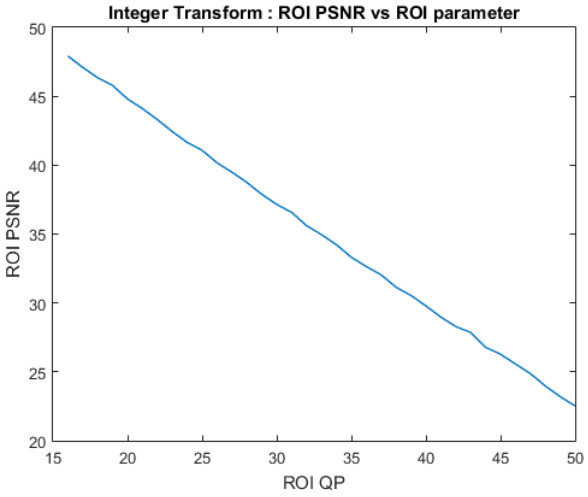
DCT based Compression

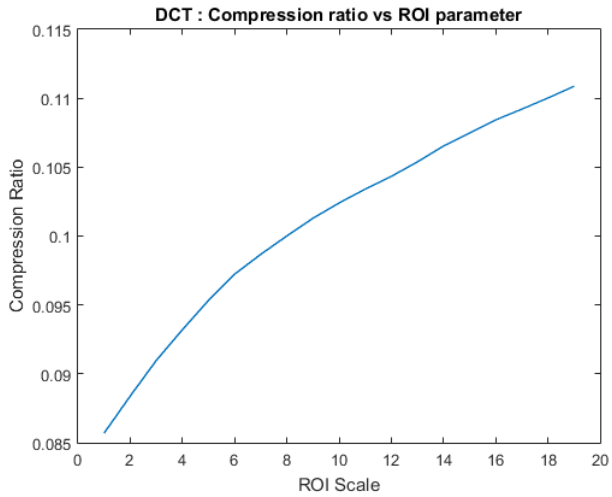
(6)

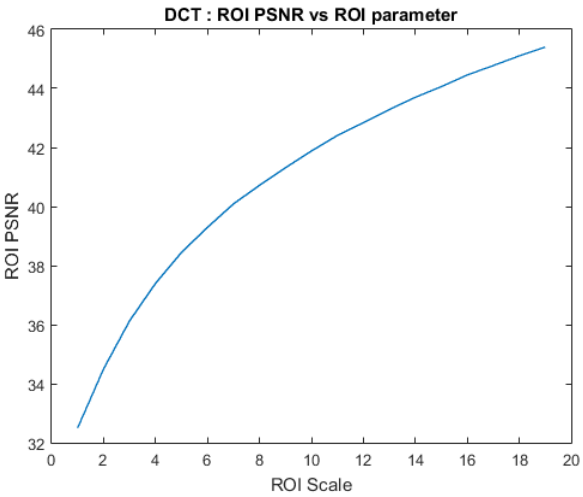
Integer Tranform based Compression

(6)









Plots showing how compression ratio and PSNR varies for ROI region for DCT and Integer Transform

The above plots shows that there is almost linear dependence between ROI and PSNR in Integer transform whereas DCT varies close to logarithmically. Also we can see the inverse dependence on ROI for compression ratio in case of Integer transform.

III. Motion Compensation.

Motion Compensation using inter frame prediction is an integral part of any video compression. Since we quantize the ROI and non regions using different measures, it will be interesting to see how those parameters will affect the P frames.

In my experiments, I have used a typical mpeg motion compensation but only used P (Predicted Frame) but not B (Bi-Predicted Frame) along with I (Intra Coded) Frame. So my GOP (Group of pictures) will look like I P I P I P I P….

Also, the residue image is generally transform and quantized the same way as I frame. But in our case, since we have ROI and non ROI, we can do the transform and quantization for Residue based on ROI scheme. But it is observed that, it did not make much difference when we use ROI scheme for Residue image compared to uniform quantization of residue image. This is due to the fact that the PSNR of P frames depends mostly on the way I frames are quantized and the residue adds a very small improvement to it. So for the following experiments we can assume, the Residue is quantized based on ROI scheme for both Integer transform and DCT.

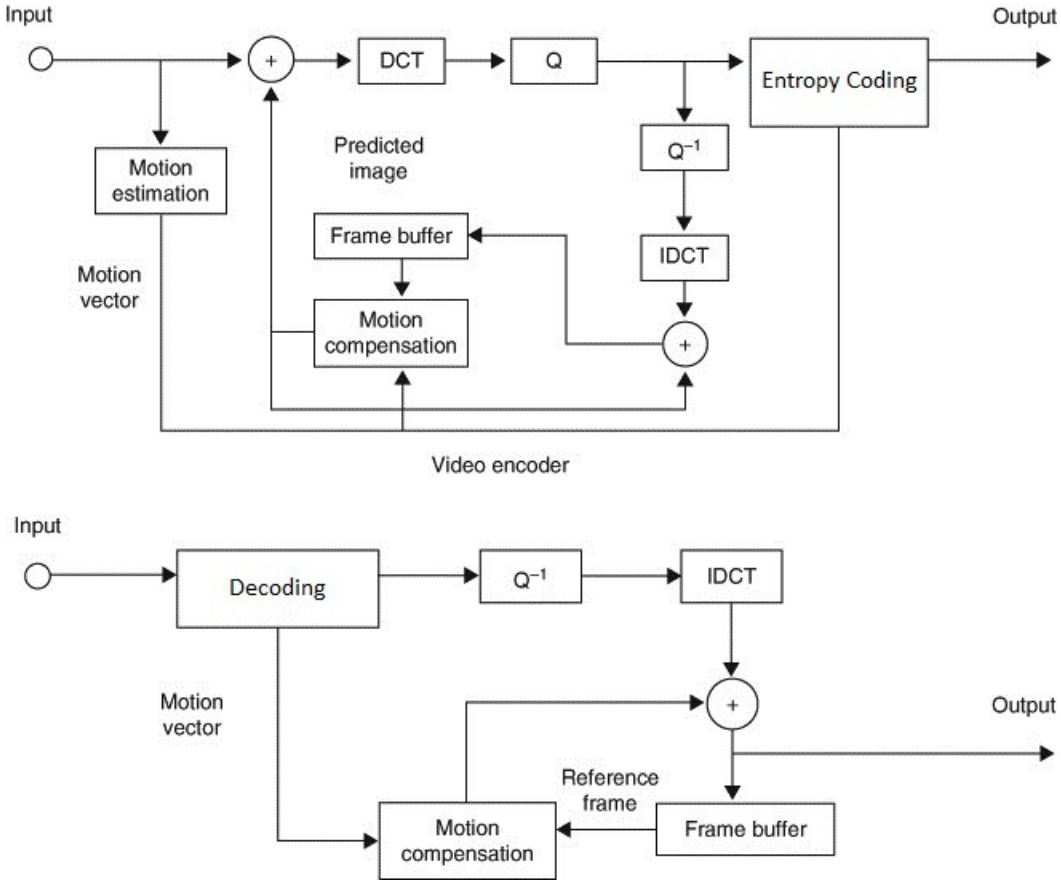


FIGURE 1MOTION COMPRENSATION IN MPEG 2

It has to be noted that in typical Video Compression one of the entropy coding techniques like Huffman Coding with Run length coding, Context-Adaptive binary arithmetic coding (CABAC) , Context-Adaptive variable length coding (CAVLC), Exponential-Golomb Coding, etc. is used. There are techniques [3] which involve adaptive coding based on ROI. This is not in the scope of this project. I concentrated on the quantization stage and motion compensation stage and thus the bit rates mentioned in this report are all based on entropy values. This gives a more accurate comparison of the techniques under study for transform and quantization stage because we need not worry about the effects of coding techniques that will come into picture if we compute bitrates based on bit stream generated by entropy coding stage.

Parameters used for next experiment with Motion Compensation [lesser no. of frames]:
Input Video : Missa (352 x 288) | 30 frames @ 15fps
Motion Vectors are predicted with a window size of 7 pixels max.

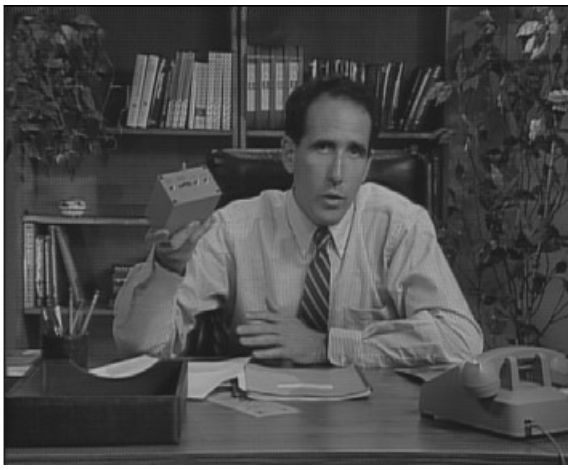
TABLE 2: MOTION COMPENSATION (30 FRAMES) - VARIATION OF PSNR WITH MACROBLOCK SIZE FOR ROI AND NON ROI

	ROI Priority Parameters	Macroblock Size	Bitrate Achieved	Video Size – Compression ratio	ROI PSNR	Non ROI PSNR	Total PSNR
1	ROI = 20 | Non ROI = 35 (Integer Transform)	16 x 16	1.2893 Mbps	0.068043	20.508	18.589	18.970
2	ROI = 20 | Non ROI = 35 (Integer Transform)	8x8	1.2865 Mbps	0.067894	20.627	18.627	19.038
3	ROI = 20| Non ROI = 35 (Integer Transform)	4x4	1.3147 Mbps	0.069385	20.901	18.671	19.157
4	ROI – 1.05 | Non ROI – 0.06 (DCT)	16x16	0.9905 Mbps	0.052279	20.511	18.462	18.883
5	ROI – 1.05 | Non ROI – 0.06 (DCT)	8x8	0.9939 Mbps	0.052454	20.6294	18.484	18.938
6	ROI – 1.05 | Non ROI – 0.06 (DCT)	4x4	1.0554 Mbps	0.055702	20.8993	18.523	19.048

Observations:

1. For the first three experiments in the above table, we use integer transform with fixed ROI and non ROI priority parameters. We can see that as the macro block size increases, the total PSNR increases.
2. The experiments 4,5,6 are done with DCT. The ROI and non ROI priority parameters are fixed for 16 x 16 macroblock size, so that all the PSNR matches with that of Integer transform. We can see that DCT performs better in compression for the fixed PSNR when compared to Integer transform for the same macroblock size.

Missa (Used in Table 2) : Smooth non ROI



Salesman (Used in Table 3) : Cluttered non ROI

Parameters used for next experiment with Motion Compensation [higher no. of frames]:
Input Video: Salesman (352 x 288) | 440 frames @ 30fps
Motion Vectors are predicted with a window size of 7 pixels max.

TABLE 3: MOTION COMPENSATION (440 FRAMES) - VARIATION OF PSNR WITH MACROBLOCK SIZE FOR ROI AND NON ROI

	ROI Priority Parameters	Macrob lock Size	Bitrate Achieved	Video Size – Compression ratio	ROI PSNR	Non ROI PSNR	Total PSNR
1	ROI = 24 | Non ROI = 40 (Integer Transform)	16 x 16	0.999 Mbps	0.049269	20.508	18.589	18.970
2	ROI – 1.05 | Non ROI – 0.1 (DCT)	16x16	1.061 Mbps	0.052336	19.508	15.521	15.882
3	ROI – 0.78 | Non ROI – 0.078 (DCT)	16x16	0.842 Mbps	0.041565	18.404	14.981	15.300

Observations:
As the number of frames increased, Integer transform performs better than DCT. This is an interesting observation. Because when we compared DCT vs. Integer Transform in Table 2, DCT seems to perform better. There, we ran the motion compensation for only 30 frames on a video (Missa) with very smooth non ROI. Here in Table 3, as we ran for 440 frames on a video with a cluttered background (Salesman) Integer transform seems to perform better.

This can be due to the fact that Integer transform is reversible, which means we can get back the exact input after doing inverse Integer transform. But this is not the case with DCT because due to

rounding off to nearest integers after quantization. So, while doing Motion prediction, we do inverse transform and quantization. So, there is an inherent error even in the Encoder side for DCT. So Motion vectors computed from this inverse DCT image will be more error prone compared to Integer transform.

Since the above experiment on motion compensation is done using a window size of maximum 7, we can expect different results for different window size..

Summary of the report:

- We used one of the best available algorithms to detect human face in real time and studied the rate and distortions for the ROI and non ROI using DCT system and Integer transform system.
- From the Transform, Quantization and Motion Compensation stages of ROI based Video Compression, experimental results shows that DCT based system performs good in Transform and Quantization stage, and Integer transform & DCT performs has their own good performing regions. For a longer video sequence with more cluttered background, Integer transform is preferred, which will be the actual scenario in real world most of the times.

Source Code:

My Matlab implementation is available in the following link:

https://github.com/pradeepkumarid/ROI_VideoCoding

A live demo based on video streamed from webcam can be done by running the standalone file: my_capture_video.m. This demo doesn't involve motion compensation.

Check Int_motion_comp_ROI.m for Integer transform based motion compensation and DCT_motion_comp_ROI.m for DCT based motion compensation.

Kindly check the readme file for more details.

Conclusion:

We saw how ROI based compression techniques can be used in Quantization stage and Motion Compensation stage. These techniques showed good results in experiments for compression without much loss of perceivable quality. Apart from doing ROI based tweaking of parameters in quantization stage & Motion compensation stage, there are techniques which allows even the coding stage to be done based on ROI[3]. The combo of all such techniques can make a big difference in handling conversational video communications.

References:

[1] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” IEEE Trans. Image Process., vol. 13, no. 10, pp. 1304–1318, Oct. 2004.

[2] C. W. Tang, “Spatiotemporal visual considerations,” IEEE Trans. Multimedia, vol. 9, no. 2, pp. 231–238, Feb. 2007

[3] Y. Liu, Z. G. Li, and Y. C. Soh, “Region-of-interest based resource allocation for conversational video communication of H.264/AVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 1, pp. 134–139, Jan. 2008.

[4] P. Viola and M. Jones, “Robust real-time face detection,” Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, 2004.

[5] Mai Xu; Xin Deng; Shengxi Li; Zulin Wang, "Region-of-Interest Based Conversational HEVC Coding with Hierarchical Perception Model of Face," Selected Topics in Signal Processing, IEEE Journal of , vol.8, no.3, pp.475,489, June 2014.

[6] Ian E. Richardson , “The H.264 Advanced Video Compression Standard” , Wiley, 2nd Edition.

[7] John Wiseman, “An Introduction to MPEG Video Compression”, http://old.siggraph.org/education/materials/HyperGraph/video/mpeg/