RESEARCH ON VIDEO OBJECT PLANE WITH APPLICATION IN TELEOPERATIONS

By

Mohsin Khan

Submitted in partial fulfilment of the
requirements for the degree of
Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
April 2013

DALHOUSIE UNIVERSITY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "RESEARCH ON VIDEO OBJECT PLANE WITH APPLICATION IN TELEOPERATIONS" by Mohsin Khan in partial fulfilment of the requirements for the degree of Master of Applied Science.

Dated: April 23rd, 2013

Supervisor: ______________________________

Readers: ______________________________

____

DALHOUSIE UNIVERSITY

DATE: April 23rd, 2013

AUTHOR: Mohsin Khan

TITLE: RESEARCH ON VIDEO OBJECT PLANE WITH APPLICATION IN TELEOPERATIONS

DEPARTMENT OR SCHOOL: Department of Electrical and Computer Engineering

DEGREE: M.A.Sc. CONVOCATION: October YEAR: 2013

____

Signature of Author

Contents

List of Tables

List of Figures

Abstract

Teleoperations is a significant field in robotics research; its applications range from emergency rooms in hospitals to space station orbiting the Earth to Mars rovers scavenging the red planet for microscopic life. We have developed a new user defined selective video object plane scheme. This selective filter works with standard H.264 encoder which is developed using Intel IPP and uses the latest multicore capabilities of new processors and can encode and transmit high definition videos over internet in real time. The area of interest is extracted and encoded at a different frame rate and noise level than rest of the frame. Our modified algorithm uses user input as well as motion detection of individual pixels to define video object plane. Video object plane filter is designed to be used for video with slow moving objects for cases like surgical procedures. The results of our compression algorithm have been verified using SSIM, PSNR and human perception survey. All these results of our VOP showed better performance than comparable encoders at the same bandwidth.

Chapter 1: Introduction

In this chapter we will discuss the motivation behind this project and how teleoperation is playing an important role in our lives.

1.1 Teleoperations

Teleoperation is a field of robotics where a robot is controlled from a distance. This distance can be as small as few feet [1] meaning that the slave robot and operator are both in same room or it can be thousands of kilometers, as is the case with space robots. A typical teleoperation system consists of a slave robot, communication channel and a master controller. Master controller in most of the cases is operated by human [2]. Teleoperated robots find applications in fields like space robots (figure 1), bomb disposal robots, surgical robots (figure 2) and hazardous waste removal robots. Operating a robot remotely in dangerous environments has a lot advantages due to the fact that the operator can work from safe distance and avoid any unnecessary dangers.

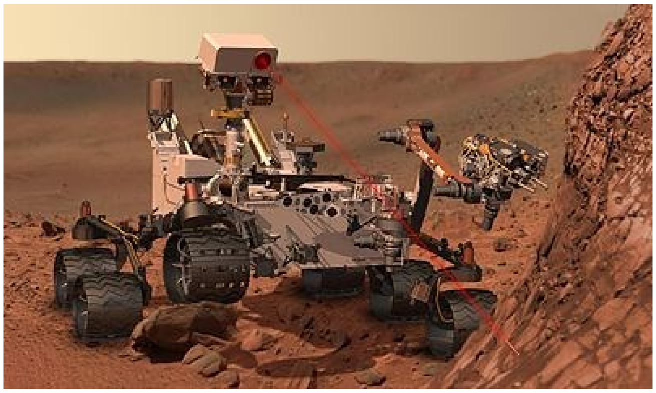


Figure 1: Mars rover (NASA/JPL)

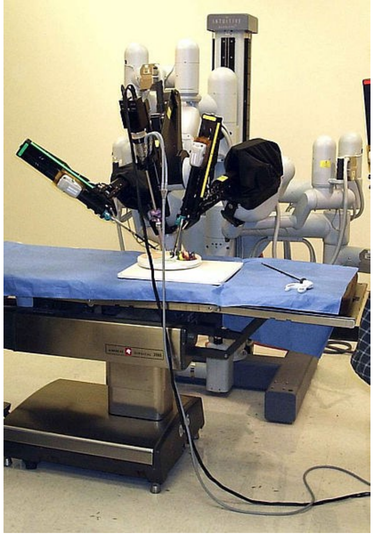


Figure 2: Surgical robot (da Vinci, Intuitive Surgical)

Any teleoperation system consists of various parts or modules. These are the slave side robot, a wired or wireless connection, a force feedback, joystick or other haptic device and master side interface. Our thesis deals with slave side control, video feedback and master side control device.

1.2 Motivation and Contribution

An easy to deploy teleoperation system with high definition video feedback is an emerging field in robotics. This thesis deals with both of these issues and we propose a novel approach of using the latest codec for video encoding. Our approach from the start has been to use already existing tools and modify them for teleoperation tasks. Another contribution of this thesis is the use of an over the shelf game controller for master side control which makes it far cheaper option to implement. In addition to it we have kept different parts of whole system separate (control system, joystick and video feedback) which in itself makes our approach easy to use and modify according to need and can be modified using different programming languages.

This thesis is focused on developing a teleoperation system which consists of a robotic arm, game controller interface for master side and an H.264 video encoder with selective filter.

1.3 Thesis Organization

This thesis is organized in different chapters. Chapter 2 details the development of teleoperation system in last three decades as well as the background of impedance control. Chapter 3 discusses the impedance control method and how the optimal solution provides an optimized solution to impedance selection. Chapter 4 is on use of game controller and how the various inputs are mapped on to the simulator to use it as a viable input device. Chapter 5 is introduction and development of video encoder, its use in teleoperation and the contributions we have done in this field. Chapter 6 discusses the results of our video encoder while at the end we conclude our research while providing future direction of this research.

Chapter 2: Background

Contact tasks are common in our daily lives. From drilling to grinding, all these tasks make use of the fact that direct force can be applied to the object. In this section we will see how the relation between force and position plays an important role in impedance control development.

2.1 Human Muscles and Contact Tasks

Humans use muscles in their arms and legs to control movement and exert force. If we look closely, it's the relaxation or contraction of muscles which produces this movement and the resulting force. With experience humans develop the sense of how much force is required to perform a specific task. The movement of muscles can be considered as change in impedance of muscles to do a specific task. To perform any task which required interaction with environment, both position and force have to be controlled. The question in the context of human being is simple to answer but in case of robots, one can ask: What happens when a robot needs to do the task which requires interaction with environment like grinding, polishing, drilling and walking? There are two answers to this question, develop accurate and robust force sensors or devise a control strategy which can provide robust control to such tasks and which can also take environment dynamics into consideration [1]. Such control scheme should be able to control both force and position simultaneously, simply saying there exists a duality property between position and force control. The problem here is that one cannot control both position and force at the same time (this is same as one cannot control voltage and current across a resistor at the same time). This relationship is based on constraints set by the environment which can be either, Natural constraints (due to the environment and robotic tool tip) or artificial constraints (due to the task to be performed).

2.2 Hybrid Control

Most of the tasks like the above have been addressed using force control, according to [2]; these force control strategies can be divided into two categories – Type I and Type II. Type I controls tends to setup a relationship between the force applied the position of the end effecter and includes stiffness control, damping control and impedance control while type II includes resolved acceleration control and hybrid position and force control both of which aim at controlling the force and position in non-contradictory ways. The proposed hybrid position and force control [1] was the obvious choice for these tasks but it had problems when grinding or cutting task were on uneven tasks [3]. For application like grinding or cutting both commanded force and position should have a relation between each other to accurately carry out the task [4]. This is where the impedance control has its application.

2.3 Impedance Control

Impedance control specifies the relationship needed between force and position rather than just specifying the quantity itself [2]. Impedance control has roots in damping and stiffness control (basic types of Type I force control). In impedance control, environment plays an important role and performance depends on environment and the value of target impedance (force vs. position) taken. Different researchers have used different techniques to find this relationship between the force and position. The basic problem with most of them (starting from the works by Hogan [4] and Craig[1]), is that these only offer an adaptive strategy for controlling the target impedance. Most of the researchers [5],[6], have used stiffness and damping relationships based on the environment dynamics. These strategies, though easy to implement and work with, do not specify whether the parameters are optimal or not.

2.4 Problems in Contact Tasks

Problems affecting the controller design for the tasks like grinding and welding are: presence of non-linear forces like centripetal and coriolis forces. These forces require an inflexible controller capable of handling non-linear forces. If we are to use linear controller here, it will neglect the non-linear forces associated with the robot motion. The situation here will degrade very fast if the task to be performed is fast because the coriolis and centripetal force increase as the square of the speed of the manipulator.

Impedance control finds its application in fields like grinding, collision avoidance [4], space robotics and biped robots [5] or in simple words where the interaction with the environment is of basic concern [6] though Hogan used impedance control for path planning which shows a special case of environment dynamics modeled as *frictionless*.

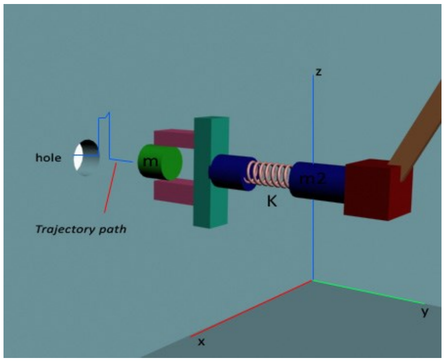


Figure 3: Famous implementation of Impedance control on peg in a hole task.

2.5 Applications of Impedance Control

Serious work on robotic manipulators and their control started in late 1970's. Initial works emphasized in force feedback or position control of manipulators. Need for a robust controller was first described by J. Craig [1] in 1981. He proposed that an adequate controller is more important than development of wrist mounted sensors or force sensors and their computation. He published his paper on hybrid control theory combining both position and force control in his paper. Though a breakthrough in control theory but had two main drawback – Firstly, the transformation of task space into joint space. This transformation is time consuming and can cause kinematic singularities [14]. Secondly he failed to take onto consideration the manipulator impedance. But [1] were successful in keeping force subspace and position subspace separate – a deal which was later used by Anderson et al. [15]. Many authors like [3] consider N. Hogan as the first man to propose the impedance control. Surely it was Hogan in 1985, who published his three papers on impedance control [4]. Interestingly, though impedance control is considered to work when manipulator is in contact with the environment, but in his third paper he used it for path planning, which is essentially a case when the frictionless environment is considered. Craig's proposal was further developed by Hogan [4], who proposed that impedance controller can be the ideal candidate for such tasks. His argument is based in the fact that impedance of manipulator should be task dependent and the relation between the interface force and imposed motion should be the impedance specified for manipulator. Mathematically;

$$Z_t = pY \qquad (1)$$

Where Z_t is the manipulator impedance, p being the weighing coefficient specifying allowable tradeoff between interface forces and motion errors. While Y is the admittance of the environment.

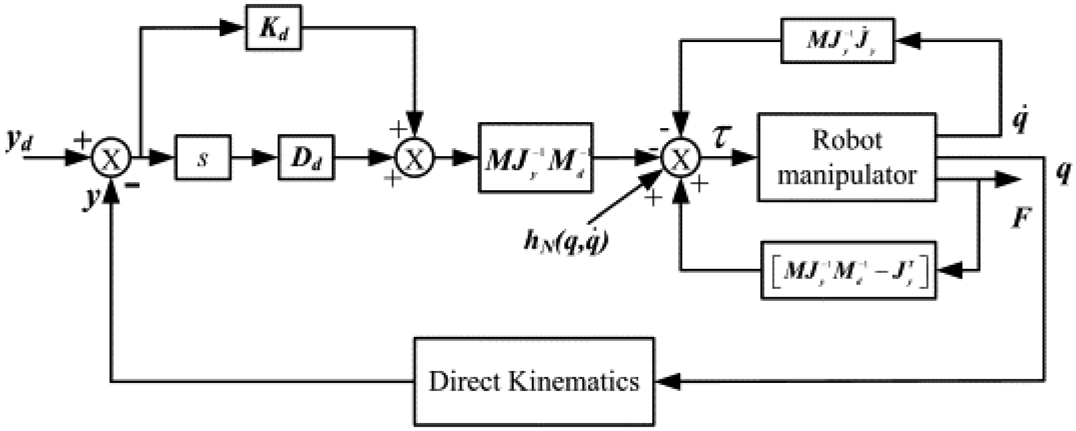


Figure 4: Impedance control simplified [16]

The assumption made it possible to use additive property of impedance even when all or some of the component of the manipulator were non-linear. A generalized impedance control based on [4], [16] is shown in figure 4.

It was not until 1988, that a more adaptable control strategy was proposed. Anderson et al [15] put forward hybrid impedance control strategy. In their paper they proposed that at any time the manipulator should behave as the dual of the environment which was later used by [8] to develop an optimal controller.

Chapter 3: Controller Development

The impedance control schemes which have been discussed work for different situation but the question is whether these are optimal or not? In addition to it uncertainties in designing of the system model are serious drawback of these schemes. To come up with an optimal controller, first serious work in this field using optimization theory and HJB equations was done by Johansson et al. [7] and later by Zaad [8].

3.1 Theory:

The idea behind the optimal controller design is to first form a performance index (cost function) of the form

$$J=½\,\tilde{x}^{T}(t_f)\alpha(t_f)\,\tilde{x}(t_f)+\int_{t0}^{tf} ½\,\tilde{x}^{T}(t)Q\,\tilde{x}\,(t)+u^{T}(t)Ru(t)dt \tag{2}$$

Matrix α is unknown and only final value of this matrix is known. Time dependent Matrices Q and R are taken as to realize the performance objective. For local stability, Q must be positive-semi definite while R should be positive definite. Matrices Q and R weigh the compromise between position error and force error [17]. The structure of Q is $\begin{bmatrix} Q11 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Qnn \end{bmatrix}$, the last term ($Q_{nn}$) acts as a penalty for the velocity of the manipulator. Since for all impedance control strategies, small motion is considered, so effectively this term can be put equal to zero. Once the above parameters are set, a Riccati equation is formed to find state dependent matrix which is further used to form an optimal control law. This general scheme of finding the control law is used by both [7][8].

Difference between the approaches is that Zaad used the optimization to find the optimized target impedance which can be tuned depending on the stiffness & dampness of the environment. On the other hand Johannson developed a controller completely based on torque. Second major difference is that [8] chose the errors in force and position as its performance indices in comparison to use of Lyapunov's stability criteria to prove asymptotic stability by Johansson et al.

As discussed earlier, Anderson's [15] approach to define robot and environment as dual of each other are central to the optimal impedance control strategy as well as other impedance controllers (though not used by[5], [18], [19]. His paper on impedance control can be summarized in one sentence – "*The manipulator should be controlled to respond as the dual of the environment*". In simple words, a dual for an inertial system is a capacitive system and vice versa. While a resistive system is considered as dual of itself. As discussed in introduction, Mathematically we can write:

$$f - f_d = Z_t(x - x_d) \tag{3}$$

The relationship here is the target impedance Z_t . Where Z_t is defined by equation (1). Like the Ohm law in electricity (where one cannot regulate both voltage and current at the same time), it is impossible to control both position and force simultaneously. Same is true for the error in both. The approach here is to minimize the sum of errors in both position and force [2][8] – a concept implemented by various techniques [3] in addition to dynamic programming. Therefore the cost function is also defined on the basis of errors (namely the position error $\tilde{x} = x\text{-}x_d$ and control (force error) $u = f\text{-}f_d$) . The cost function defined by both [7][8] is of the same form.

In general, the cost function is defined as

$$J=½\, \tilde{x}^T(t_f)\alpha(t_f)\, \tilde{x}(t_f)+ \int_{t0}^{tf} ½\, \tilde{x}^T(t)Q\, \tilde{x}\, (t)+u^T(t)Ru(t)dt \tag{4}$$

For which the Hamiltonian is

$$H\,(t) = ½\, (x^TQx+u^TRu)+\lambda(Ax+B_uu+B_ww) \tag{5}$$

Where the constraints are modeled by the following equations:

$$\dot{x}=Ax+B_uu+B_ww \tag{6}$$

Here A, B_u and B_w are the ratio between environment stiffness & damping, inverse of environment damping respectively. The term w is the disturbance term.

Minimizing this Hamiltonian with respect to control u, where u is force error we get a general form of control.

$$u(t) = - R^{-1} B_u^T \beta(t) - R^{-1} B_u^T \alpha (t) \tilde{\boldsymbol{x}} (t) \tag{7}$$

in these equations β(t) is the term to cancel the disturbance effects (not considered by Johansson because of assumptions that external forces are constant) for which the control is

$$u(t) = - R^{-1} B_u^T \alpha (t) \tilde{\boldsymbol{x}} (t) \tag{8}$$

α(t) here can be found using K-Method as discusses by [17]. So, the equation to find is a first order system of differential equations with final value given. The eq. is

$$\dot{\alpha} = \alpha B_u R^{-1} B_u^T \alpha - \alpha A - A^T \alpha - Q \tag{9}$$

Both β(t) and α (t)are calculated back in time because final values are known. Important point to note here is that Zaad calculated the value of β(t) and α(t) to find the commanded position (and used a position controller scheme) while Johannson used the same form of control variable to find torque to be used in following equation:

$$\mathrm{M}(x)\, \ddot{x} + \mathrm{C}(x, \dot{x}\ )\, \dot{x} + \mathrm{G}(x) = \tau + \lambda\, (x)^{\mathrm{T}}\, \mathrm{F} \tag{10}$$

Where x= position coordinate
$\dot{x}$= velocity
$\ddot{x}$ = acceleration

While τ is the control torque. M (q) is the general moment of inertia, C is the Coriolis and centripetal force and F is the contact force at the end effecter. The reference trajectory for the manipulator is assumed to be available as functions of time. The objective here is then to follow a given, bounded reference trajectory without position and velocity errors which are $\tilde{x}$ and $\dot{\tilde{x}}$ respectively.

Approach by Zaad et al. is more applicable than Johannson et al. (though it requires considerable computation power to solve for α which in present world is not a handicap). As commanded force and position both can used to control the manipulator as compared to just torque. This gives greater flexibility in designing controller. As discussed by Johannson [7], all impedance control schemes suffer from slow transient response. Same is true for the controller designed by Zaad. Though the controller has a slow transient but it has a lower overshoot as well as smooth response. Again as with all impedance control schemes [3] the environment plays an important role in determining the convergence point.

Another very simple impedance controller is discussed by [20]. This impedance controller is not general in a sense that it is defined for very specific task and is based on damping control. But the simplicity of the controller and its results show that minimizing a damping based cost function gives better results than other approaches and the computation needed is also less as required by previous controller design.

Chapter 4: Game Controller

For controlling the movement of robot over internet, an Xbox 360 controller is used. This controller is connected to master computer through proprietary 2.4 GHz radio connection. The controller has two analog joysticks, eight digital buttons and a four way digital control pad.

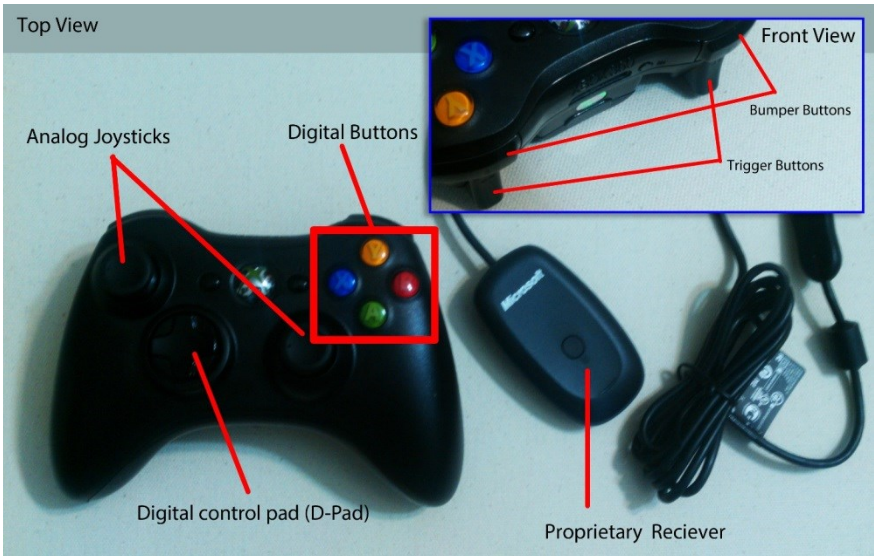


Figure 5: XBox controller and its various buttons

The XInput API (application programming interface) for Xbox controller is available through MSDN network. This API includes libraries and functions to access the controller and map the desired actions on to any program being run on computer. The four major functions of API which are used in our software are as follows: [23]

Function	Description
XInputEnable	Turns on XInput.
XInputGetCapabilities	Triggers and recognizes available inputs of controller.
XInputGetKeystroke	Keystroke input identifier
XInputGetState	Retrieves the current state of the specified controller.

Table 1: Xbox API functions

4.1 Interfacing with Software

Xbox controller needs a radio receiver to connect to computer. Once the controller is connected, the software automatically recognizes it and waits for input from it. XINPUT_GAMEPAD structure is used to read values from keystrokes. The bitmask of each digital buttons is shown in table below (these bitmasks are provided by [23]):

Device button	Bitmask
XINPUT_GAMEPAD_DPAD_UP	0x0001
XINPUT_GAMEPAD_DPAD_DOWN	0x0002
XINPUT_GAMEPAD_DPAD_LEFT	0x0004
XINPUT_GAMEPAD_DPAD_RIGHT	0x0008
XINPUT_GAMEPAD_START	0x0010
XINPUT_GAMEPAD_BACK	0x0020
XINPUT_GAMEPAD_LEFT_THUMB	0x0040
XINPUT_GAMEPAD_RIGHT_THUMB	0x0080

Device button	**Bitmask**
XINPUT_GAMEPAD_LEFT_SHOULDER	0x0100
XINPUT_GAMEPAD_RIGHT_SHOULDER	0x0200
XINPUT_GAMEPAD_A	0x1000
XINPUT_GAMEPAD_B	0x2000
XINPUT_GAMEPAD_X	0x4000
XINPUT_GAMEPAD_Y	0x8000

Table 2: Xbox bitmask values for each button

The analog inputs from the two joysticks on controller each have a value between -32768 and 32767 where 0 is the center.

4.2 Error Avoidance

In order to avoid any unintentional keystrokes the software only registers keys once the right trigger button is pressed down. In order to control the robot, the user needs to keep this key pressed. As soon as the key is released the software stops registering the strokes and ignores any kind of interaction from the controller.

Chapter 5: Video Object Plane

Video broadcast is one of the most important requirements of teleoperational system. Surgical robots [23] and mobile robots [27] require video to be relayed to master controller for situation awareness purposes. Video feedback gives the operator sense of what is going on in slave environment and based on this the operator can make decision for the next move. In recent years, availability of high-definition cameras has made it possible to shoot footage in high resolution. This high resolution imagery is clear, has more contrast ratio and gives a better picture of slave surroundings. On the downside, streaming high definition video over internet requires high bandwidth. Part of our research is focused on 'selective' compression of high definition video to conserve bandwidth. In the next sections we will see how the present day compression technologies are being used and how we have changed the existing compression algorithms to better serve our high definition on low bandwidth need.

5.1 Video compression standards

Like most other standards in telecommunication, the video compression and transmission standards are well defined by their respective bodies. The governing body for video compression standards is VCEG (Visual Coding Experts Group) which is part of ITU (ITU Telecommunication Standardization Sector) [25]. This group is responsible for H.26x line of standards for video compression. The most recent standard in VCEG compression standards is known as H.264/MPEG 4-AVC and is one of the most widely used standards for video compression today [25][26].

5.2 H.264/AVC Encoding / Decoding

H.264 standard uses various methods to compress a video and has better performance in preserving quality [26]. The most efficient method in compression is inter-frame prediction. In this method each frame is divided into macro blocks which can be used to predict next frame. Instead of encoding each frame, the encoder looks for similar macro blocks among these in a frame which was encoded previously. In cases where the whole video is available, it can even look at future frame to predict the required macro block. When the block is found which is similar to one in previous frame, it used the previously used block and thus eliminating the need to encode every pixel in a given frame. For cases where the macro block is found, but is not at the correct location, the motion vector is computed and used to alter the resultant frame. Simply defined, it is method of defining a single frame based on neighbouring frames. This inter-frame prediction method uses motion estimation (figure 6) and motion compensation between the frames [26]. A simplified version of inter frame prediction is shown in following figure 7. H.264 encoder uses previously encoded frames to predict the next frames. It can store up to 16 reference frames for this purpose. Using this technique, the H.264 performs better in applications where the motion is repetitive. Previously the reference frame size used to be 2 (one backward, one forward). This B-frame method is only applicable on recorded videos but in cases where the video is real time or live stream, the 16 reference frames give better compression.

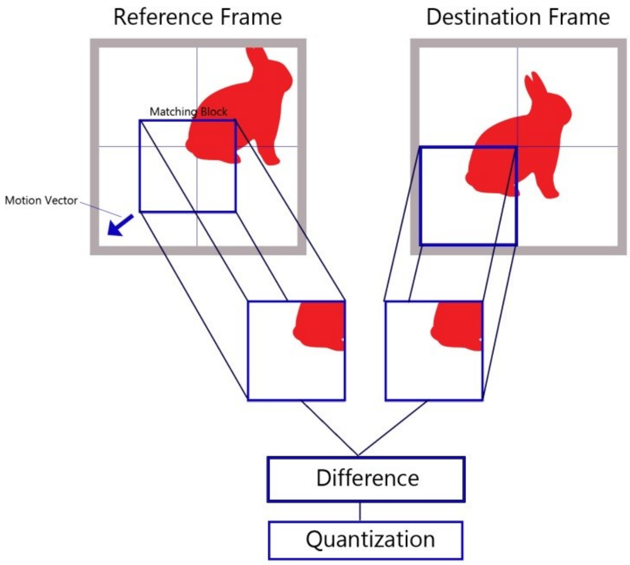


Figure 6: Motion estimation in an H.264 encoder

Earlier video broadcasting systems in robotics used commonly available webcams for streaming video. Although a simple solution, most of the webcam available do not offer high definition video, high contrast ratio and these are vulnerable to low color fidelity. The ones which offer all of the above require high bandwidth. In addition to stand alone systems, researchers have used 3rd party software to stream video including Skype ™ and other IP based camera systems [30]. These software solutions are easier to setup, allow multiple logins and updated regularly by their respective manufacturers. Drawbacks of these systems include system down times [31], sudden decrease in quality if the server becomes busy or low bandwidth is detected. The solution in this scenario is to use high definition cameras and use effective compression so that less bandwidth is used in transmitting video. Previous research [32] has shown that H.264 is suitable standard for medical imaging, thus use of this encoding scheme in teleoperated surgical robot makes a compelling case.

5.3 Video Object Plane

The problem with most of the robotics application of video is that there are no fast moving objects and most of the frame is filled with object which is of no interest to operator. After analyzing different footage of surgical processes, we found that the area of interest in one frame is roughly 30%-40%. In addition to it, in most robotic applications (specifically surgical robots) the movement of robotic arm is slow and object being operated is either still (skull surgery) or moves at relatively slow rate (heart beating – the term slow is relative here, considering a normal heart beats at 70 times per minute relative to 30 frames per second of a video).

Based on our research, we proposed that a video encoder which tracks motion of an object in a video frame and only encodes that object into high definition and high frame part of compression process can give better results.

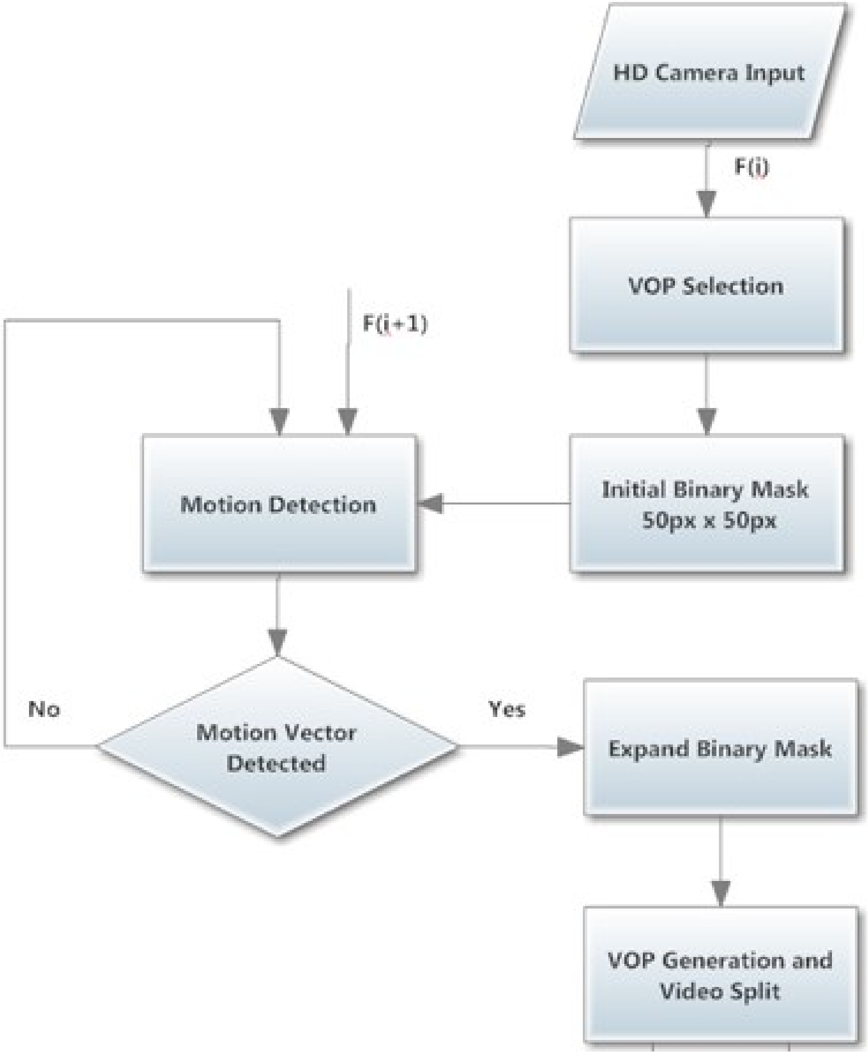


Figure 7: VOP selective filter

This was achieved by using a selective filter which applies a binary mask to video before the video is encoded. This filter separates each video frame into area of interest and the masked area. This method forces the algorithm to only use this selected part of the frame for inter frame prediction while masked part of the frame is encoded at a low frame rate and low pixel density.

The area of interest is chosen by two methods:

1. By the operator
2. By comparing two consecutive frames for motion vectors

Previous researchers [28][29] have worked on similar methods but the technique used did not give any control to operator to choose area of interest neither did it encode the rest of the area as video. The previous methods were based solely on motion estimation or intensity changes in a frame. The disadvantage of such method is that if there are more than one object which is in motion in a particular frame, that object is also encoded and this can unnecessary encode objects figure 9 which are useless to operator and thus making the whole process consume more bandwidth and processing power. In addition to it, the method used can only work in offline cases due to high computational requirements.

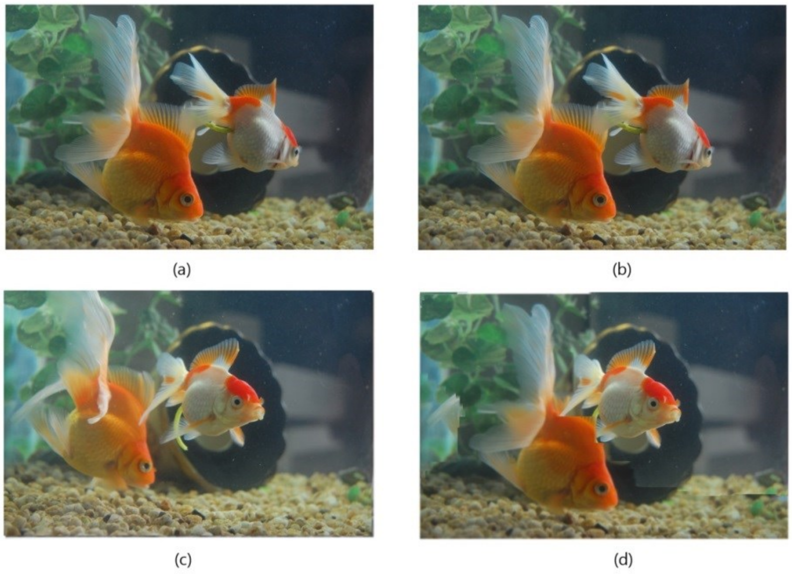


Figure 8: Comparison of two binary masks schemes (a) Our selective filter (b) [26] method. Note the second fish is treated as a static image in other researchers work.

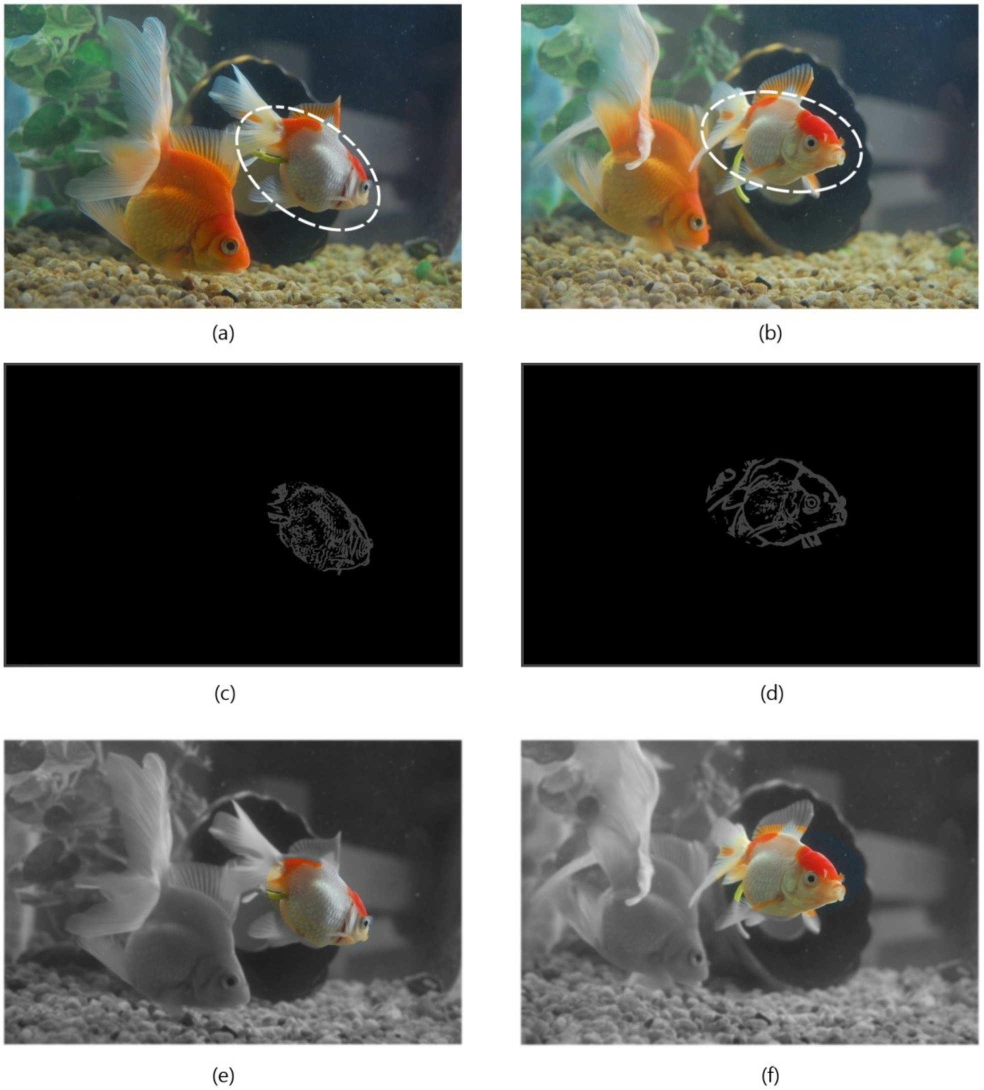


Figure 9: H.264 encoder with selective filter (a) and (b) are two frames for a video sequence shown. (c) and (d) show the selected area which is encoded at high quality (e) and (f) show the output (area outside selected area is passed through black and white filter)

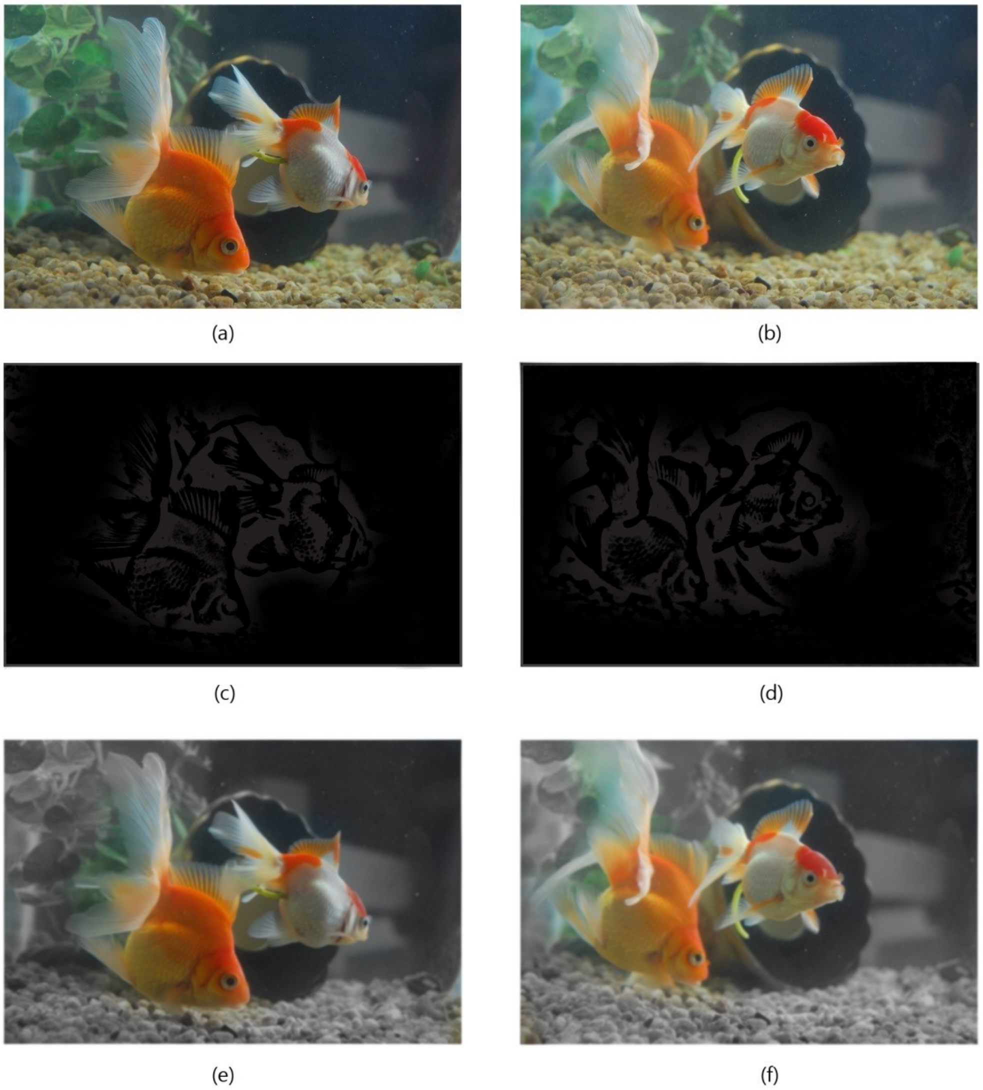


Figure 10:: H.264 encoder without selective filter (a) and (b) are two frames for a video sequence shown. (c) and (d) show the selected area which is encoded at high quality (e) and (f) show the output (area outside selected area is passed through black and white filter)

5.4 Clipping Mask or Binary Mask

The initial binary mask (which is applied to only first frame) is selected to be a circular area which is 50px in diameter around the selected point. This 50px area serves as a base for encoder and this area is encoded regardless of whether there is motion or not. Circular mask gives better coverage compared to square masks and helps reduce the unnecessary corner overlap. Initial binary mask is shown in figure 11 below.

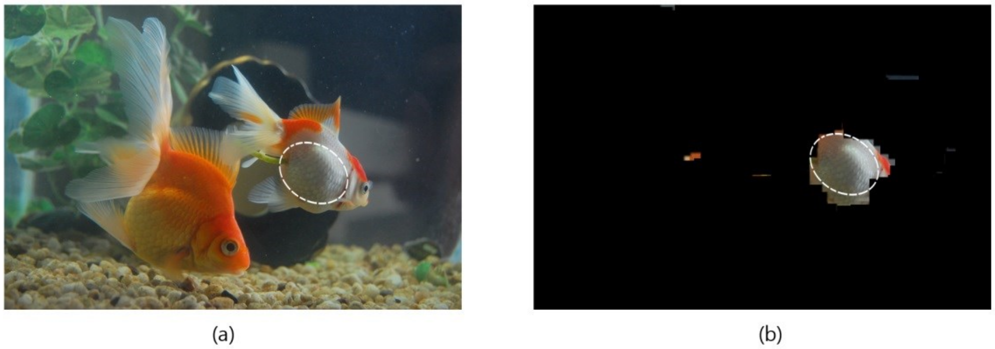


Figure 11: Initial mask selection (a). The binary mask is formed around the selected point in 50x50 px wide area (b).

The next step is to modify this clipping mask to cover the whole object. For this part motion information of pixels between consecutive frames is analyzed. Based on displaced pixel difference, algorithm forms different zones of motion in the frame. Only the zone which contains the initial clipping mask is retained while reset is discarded (figure 12b). In cases where the initial mask overlaps two or more zones, all zones are kept for encoder 1 while rest is masked and sent to encoder 2 for encoding.

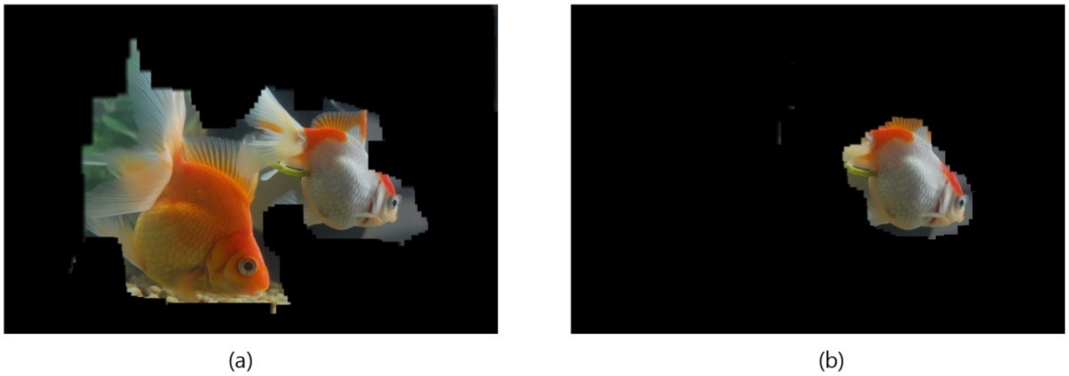


Figure 12: Binary mask modification. (a) binary mask based on motion vectors alone. (b) Binary masked after discarding area not selected by user.

5.5 Object Tracking and Fine Tuning of Mask

Once the required zone in the frame is identified, the mask is saved and the algorithm moves to next set of frames. On these frames the motion of the initial mask is tracked and the whole refined mask is moved relative to that point. In order to fine tune it further, the algorithm checks the pixels at boundary of mask for motion, if the motion vector of those pixels exceeds the threshold value, the mask is scaled up or down depending on the positive or negative value of motion vectors. A positive motion vector (away from center) adds to the binary mask while a negative vector (towards the center) decreases the size of the mask. The change and tracking of binary mask is shown in figure 13 and 14. The reference point in this frame is taken as the fish body.

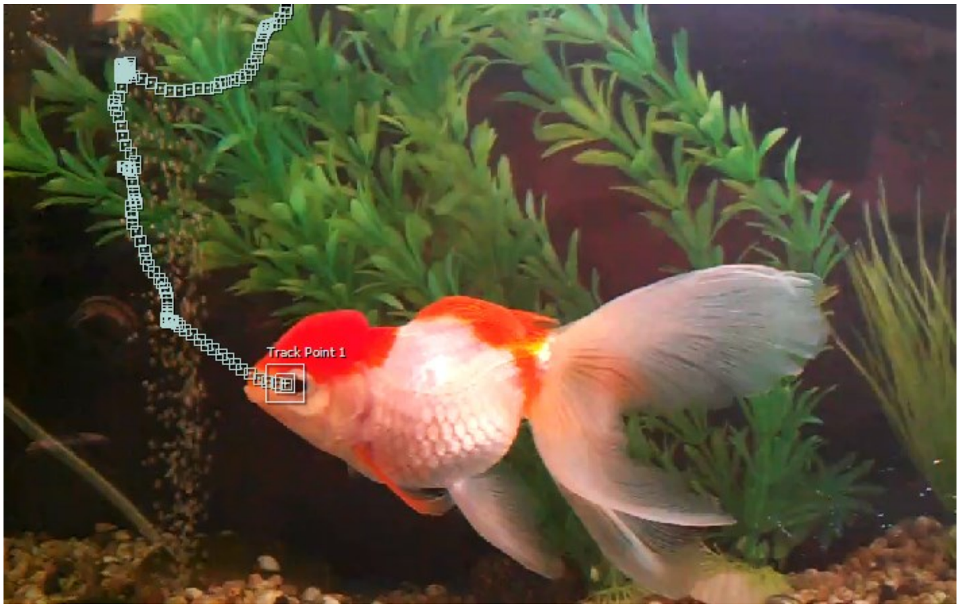


Figure 13: Movement of fish over 100 frames tracked. Only first frame with future tracking points are shown.

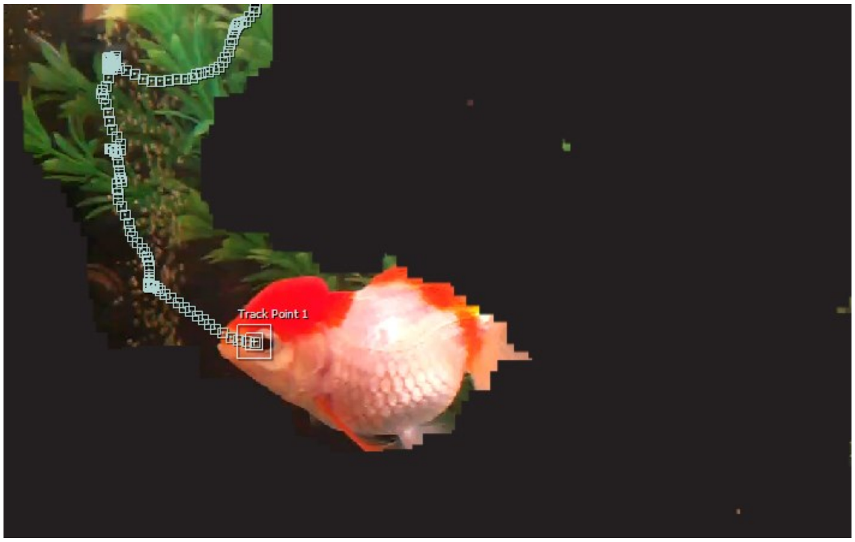


Figure 14: Mask movement following the track points. First frame and tracking point of next 100 frames is shown. The mask follows these tracking points.

Complete algorithm for our compression method is shown in figure 8. Another advantage of this method is that any video encoder can be used as the selective filter is independent of encoder. Using this technique allows the user to different encoders with the selective filter depending on the system being used. In addition to it, since the encoder is strictly written according to ITU guidelines, the master unit does not need any special decoders to decode it thus avoiding the need to of installing any new plugin or application on master computer.

5.6 Full Frame Change Detection

In case where the whole frame changes (example being robot tip moving to totally different spot or robot tipping over), the area of interest selected is discarded and whole frame is encoded as one. To check for full frame change luminous component change between two frames is calculated. If the average luminous change is more than the threshold specified (taken as 30% of total), the algorithm discards previous selection and starts to look for new area of selection. Full frame luminance component is defined as:

$$FL_k = L_{k+1} - L_k \tag{11}$$

Where L_{k+1} is the average of luminous component of k and $k+1$ frame while L_k is average luminous component of k and $k-1$ frame. Any rapid change above this threshold will automatically switch off encoder 2 and force encoder 1 to encode full frame without any selected area.

Figure 15 shows the zoomed in image. The different between the masked area and unmasked area is evident in the right side image. The area not covered by mask has less noise and more contrast to the area which is masked.

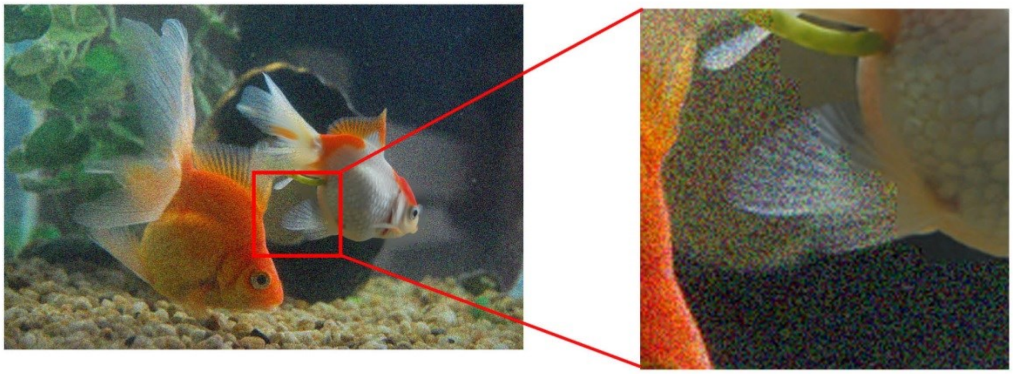


Figure 15: Zoomed Image showing the two different pixel rates of one frame

5.7 H.264 Encoder

The H.264 encoder can be implemented in two ways. Either software based or directly on hardware. For software based encoding there are different schemes and libraries available, for our encoder purpose we have used Intel IPP *(http://software.intel.com/en-us)* for designing the filter and encoder. The decision to use Intel IPP is based two reasons: firstly it has been shown by previous researchers that Intel IPP based encoder has better results compared to others [26], secondly Intel IPP gives better access to parallel processing on a multi-core processor. Since the encoder uses reference frames stored in memory, naturally the hardware based algorithms are faster and more efficient. This allows use of multicore capabilities of new processors thus decreasing the computation time for real time compression.

From the figure 16, it is evident that the whole compression scheme can be divided into two parts: initial selective filter and two H.264 encoders.

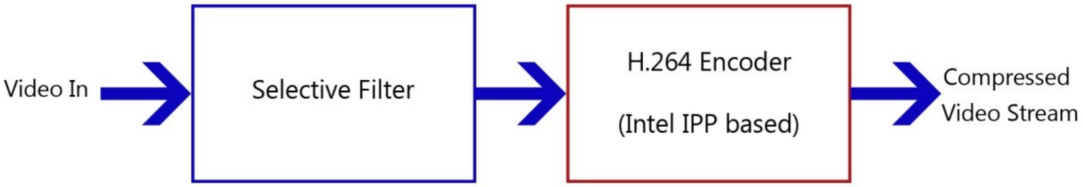


Figure 16: VOP with H.264 encoder

The filter uses both motion tracking and boundary tracking of the object at the same time. In the initial phases, the operator selects the object of interest in the frame. At this point the software looks for the boundary of the object and outlines it with white marker (this is just an extra feature and can be turned off). Once the object is identified, the motion tracking of the object begins.

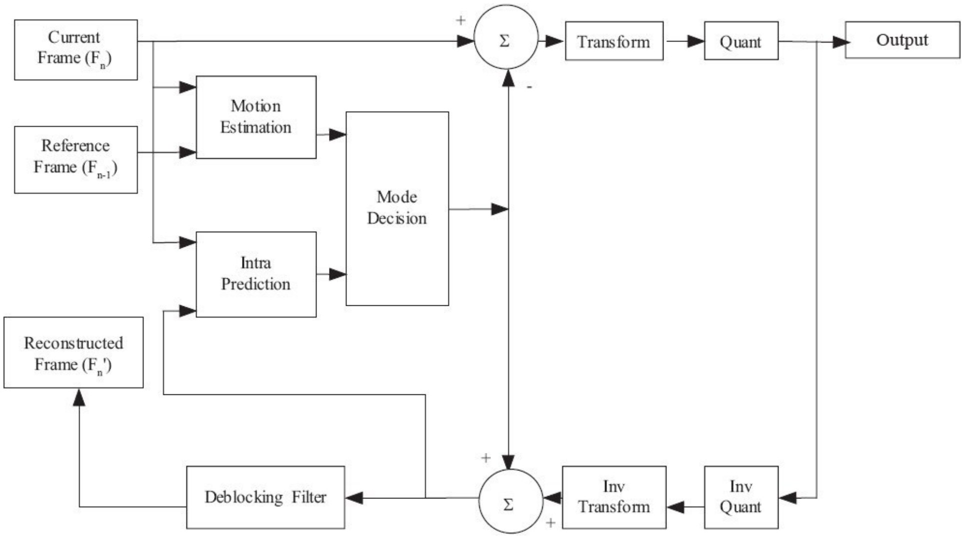


Figure 17: Simplified H.264 encoder [36]

This motion tracking is not only based on the boundary but also on the position of that object in the frame. Thus if the robotic arm changes its orientation with respect to the object, the boundary will change. This selected part of frame is then sent to the encoder 1 (sub part of main encoder), while rest of the frame is sent to encoder 2. And since in any case this selected part will be at most equal to or less than the size of complete frame, the encoder 1 will only be encoding that part thus making the encoding faster and giving us better compression. The encoder part of our compression software uses inter-frame encoding where two successive frames are compared to each other for motion estimation. If two similar blocks in two different frames are detected, only information regarding their motion is transmitted instead of complete block information. For teleoperation systems where the tool movement is slow (surgical robots or bomb disposal robots), this scheme gives relatively high compression rate with minimum effect on quality of transmission.

The video compression and video transmitter developed by us works parallel to Matlab on slave computer. This method of keeping video encoding and transmission separate from control software makes it easier to keep it cross platform compatible, change transmission settings, debug system and most importantly allows multiple users to view task space without giving them control of robot itself. And if required, the encoder can be run on a separate computer (but to master, it appears to be one slave computer). In order to make things simpler, whenever *[engage]* command is used on master computer, webcam and compression software are initiated automatically. This way, user does not need to run two or three software and can have one control panel window open to interact with both the robot and the camera. The user also has the ability to choose his connection speed to choose between different resolutions of video available. This information is relayed to encoder and transmission resolution is set according to available data connection.

5.8 Video Decoder

The encoder is written strictly according to ITU-T H.264 [25] standard recommendations; no additional decoder is required on master computer. If a user does not require our interface and only wants to view video, received video can be decoded by most modern browsers. But to use our video settings interface (in order to see control panel), transmitted video can be loaded (using *NetStream* object) and decoded by Flash 10.1 or higher without installing any additional plugins (Flash is installed on nearly 99% of computers connected to internet [35]). Our master interface is compatible with Windows, Mac OS and all tablets running Android OS v3.0/4.0. For smartphones however, it can only run on those equipped with 1 GHz or above processors and Android OS v2.3/4.0 (mostly because of high processing power needed to show flash format and lack of support on slower smartphones).

Chapter 6: Results and Discussions

In this section we will discuss the master-slave position and force results and also compare our filter-encoder scheme against already existing methods of video compression. The testing for this section was done using a Canon 60D APS-C HD camera. The decision to use this particular model was the fact that video footage can be recorded in RAW format which is uncompressed unlike other point and shoot or HD cameras where the footage is compressed on using onboard chip.

6.1 Simulation results for Master-Slave System

The use of common game controller made it difficult to measure the force feedback as the feedback is limited on such devices. Due to this limitation, we have used the position data from the controller (using Xbox SDK) and from the robot simulator. The values from the slave computer were sent to the master side and mapped on to the virtual robotic arm on the screen. Different experiments were carried out where the robotic manipulator was moved in different position and the resultant position of the arm was measured. Figure 18 shows the simple raising of the arm and then putting it down. The slave follows the master command within specific (less than 10ms) time delay. Similarly figure 19 shows hold and move of robotic arm with reasonable amount of tracking by slave to master command. Figure 20 and 21 show a quick raising and lowering of the robotic arm. The tracking data from these simulation show that though there is a delay (which is always there in teleoperational systems), but it is within allowed time delay values.

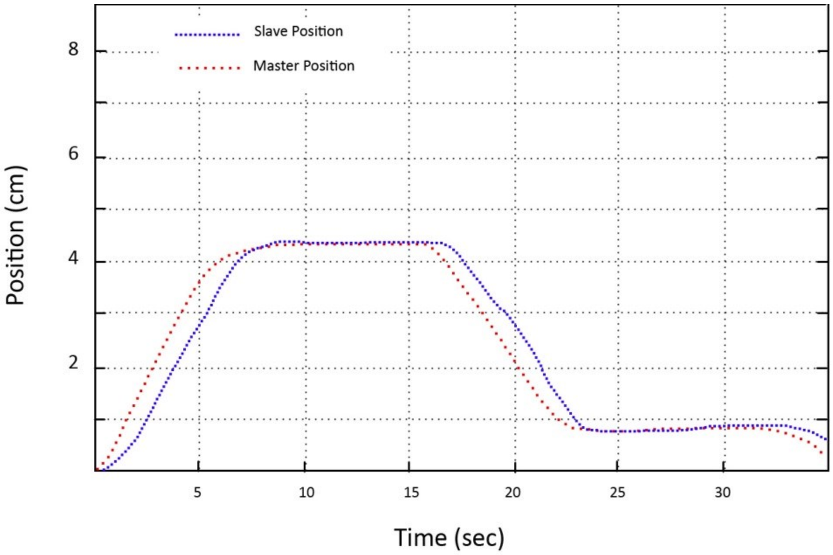


Figure 18: Position data of the master and slave

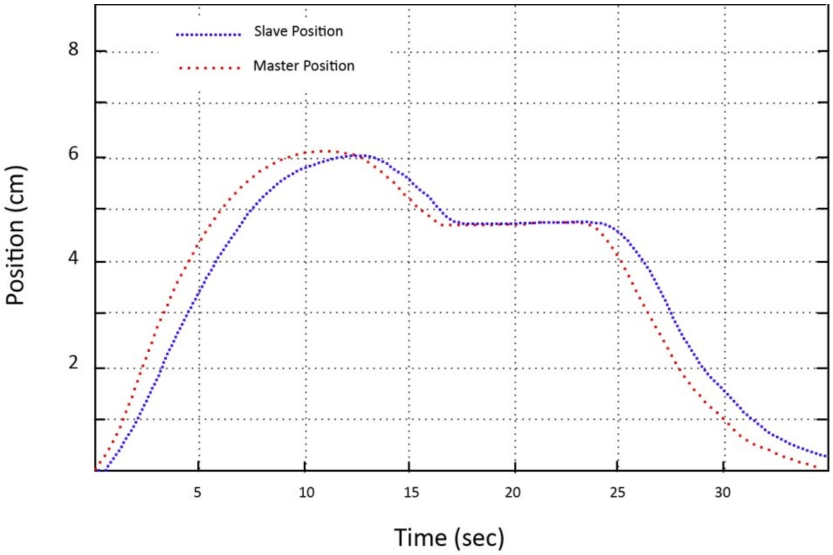


Figure 19: Position data of the master and slave

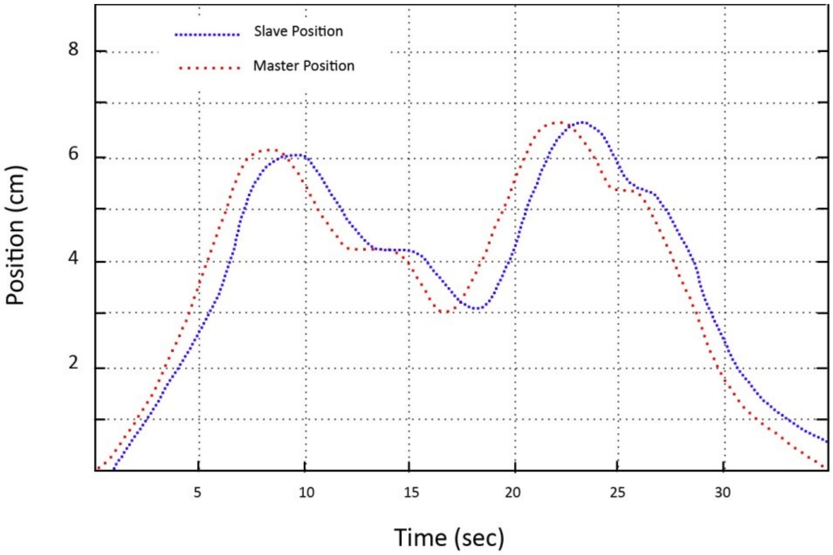


Figure 20: Position data of the master and slave

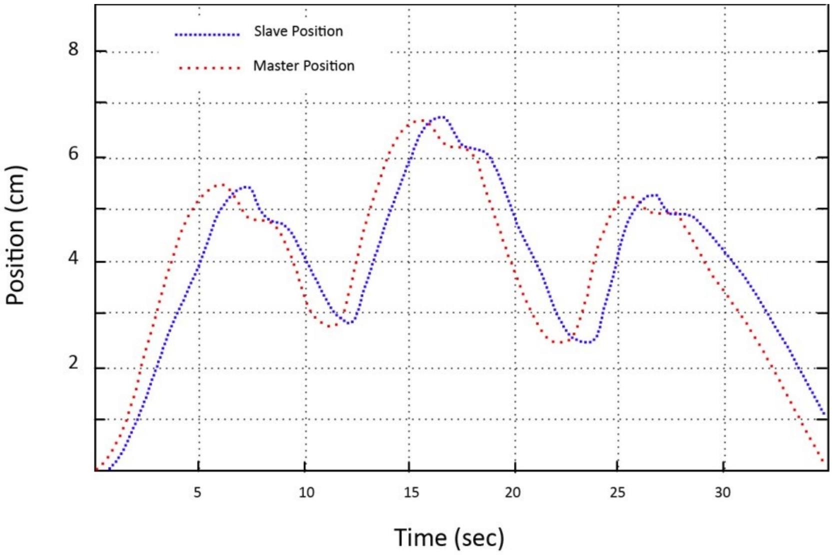


Figure 21: Position data of the master and slave

6.2 Comparison between Selective Encoder and Other Encoders

In order to benchmark our selective filter encoder, we tested it against Intel IPP (without our filter) which has been used by previous researchers [26]. In order to keep performance analysis similar to previous research, we have used the same metrics as used by [26]. These metrics are:

- Structural similarity index metric (SSIM)
- Mean square error (MSE)
- Peak to peak signal to noise ratio (PSNR)

In addition to above metrics, subjective video analysis was also performed.

Two videos were shot using standard HD camera and the video was encoded and transmitted over a local area network. Technical difficulty made it impossible to encode the same video with three encoders at the same time, to overcome this issue the same video was recorded without compression (some cameras have on-chip video compression, this option was turned off and the video recorded was in raw format) so that the other two encoders can be applied to the same frames for analysis purposes later.

6.3 Case 1: Swimming Fish

In order to compare our selective filter, a subject which moves at slow rate and has a more organic shape was needed. A short video of swimming goldfish was shot because not only it has a slow movement and organic shape but it is also unpredictable. The area of interest was selected to be the main body of the fish minus the fins. Over all the fish made up less than 25% of the frame. The 54 frames from first two seconds are shown in figure 22:

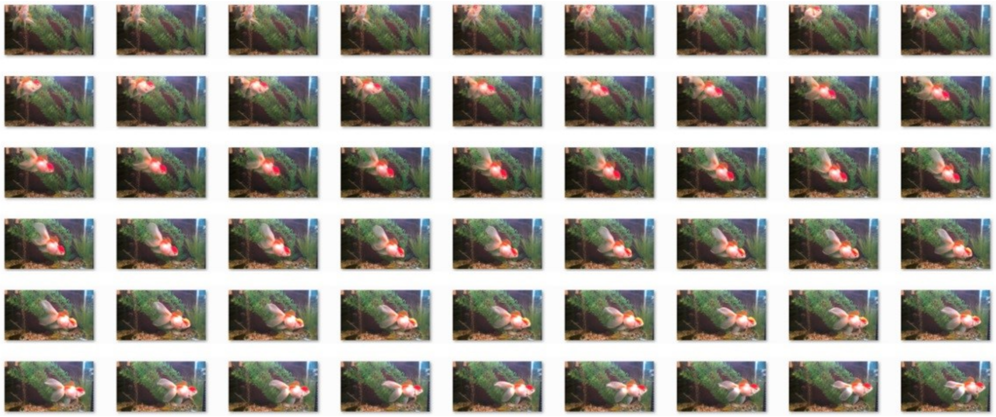


Figure 22: The video of goldfish swimming. Same subject was shot at 1080p and 720p resolution.

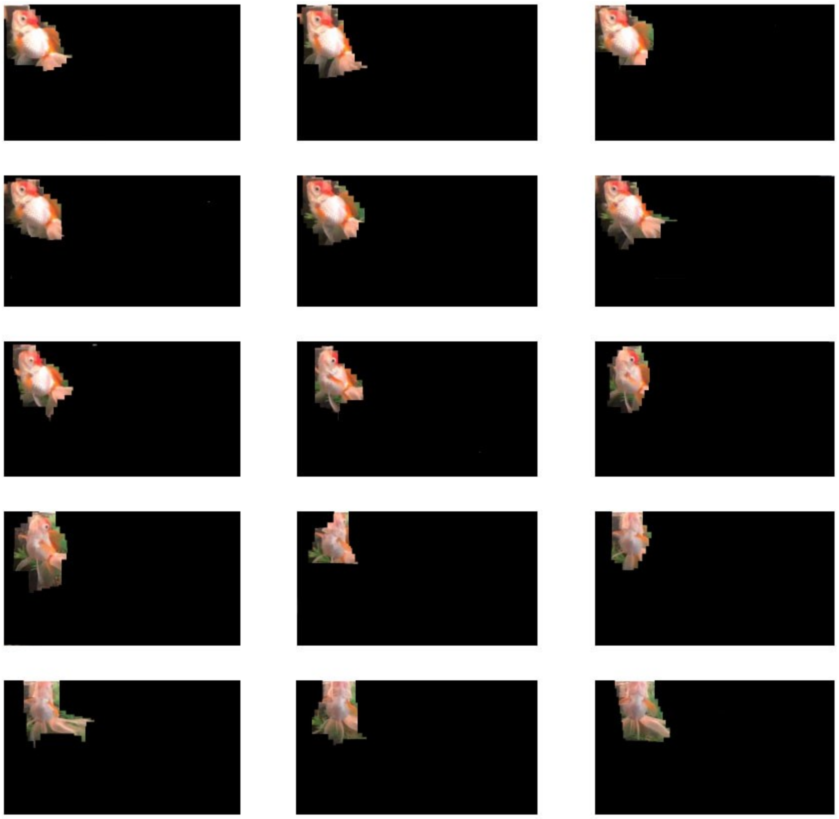


Figure 23: Binary mask on first few frames of video

The figure 23 shows tracking of binary mask. It can be seen that tracker keeps the binary mask relatively equal to organic shape of the fish. This tracking and change in shape results in better compression of desired VOP and avoids encoding unnecessary pixels from the background.

The parameters for goldfish 1080p are as follows.

Height of frame (px)	1080
Width of frame (px)	1920
Frame rate (fps)	29.97
Total size (in MB)	138
Total running time (in sec)	23

Table 3: Parameters for Goldfish 1080p

The second video was shot with same camera but the resolution was decreased to 1280x720. The parameters for goldfish 720p are:

Height of frame (px)	720
Width of frame (px)	1280
Frame rate (fps)	29.97
Total size (in MB)	97
Total running time (in sec)	23

Table 4: Parameters for Goldfish 720p

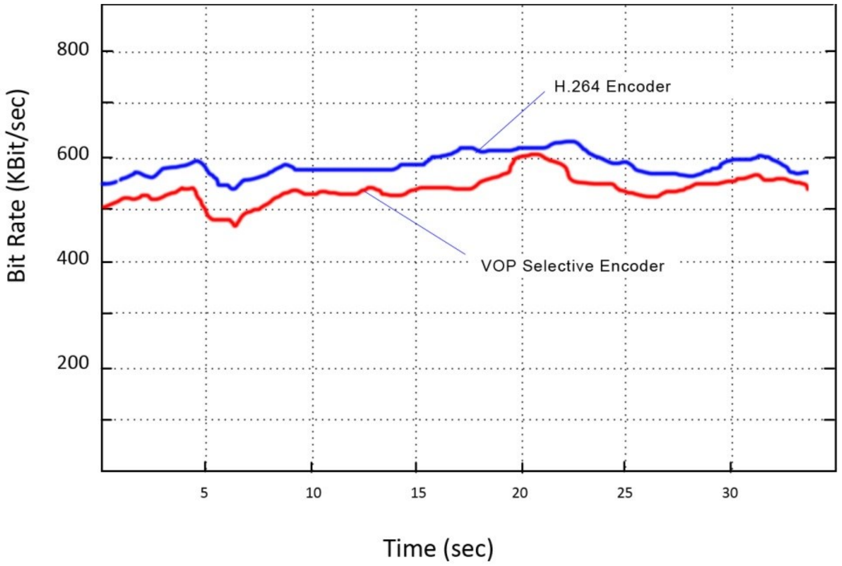


Figure 24:Bit rate of Goldfish 1080p for VOP selective filter and H.264 encoder for same quality and same PSNR

The video of goldfish was encoded at variable bit rate with constant PSNR value. Figure 24 shows lower bit rate for the VOP compared to an H.264 encoder without selective filter option.

6.4 Structural similarity index metric (SSIM)

SSIM is defined as measure of similarity between two images where one images is untouched or uncompressed [26]. SSIM has been proved to be a better way of quality analysis than MSE or PSNR [26].

SSIM is defined as:

$$SSIM\ (x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

(12)

Where μ_x and μ_y are the average of image patch x and y, σ_x and σ_y are sample deviations, σ_{xy} is the covariance and c is the constant which stabilizes the term. [26]. The value of SSIM fluctuates between -1 and 1, which is 1 only when both images are same. SSIM analyzes an image or a frame (as in our case) in 8x8 patches and complete SSIM is simply an average of complete SSIM of frame.

Due to the fact that we have two different levels of compression in each frame, SSIM was applied to overall frame as well as only on the area selected by our pre-encoder filter. In figure 25 we can see that VOP has 10% better SSIM values than an H.264 encoder.

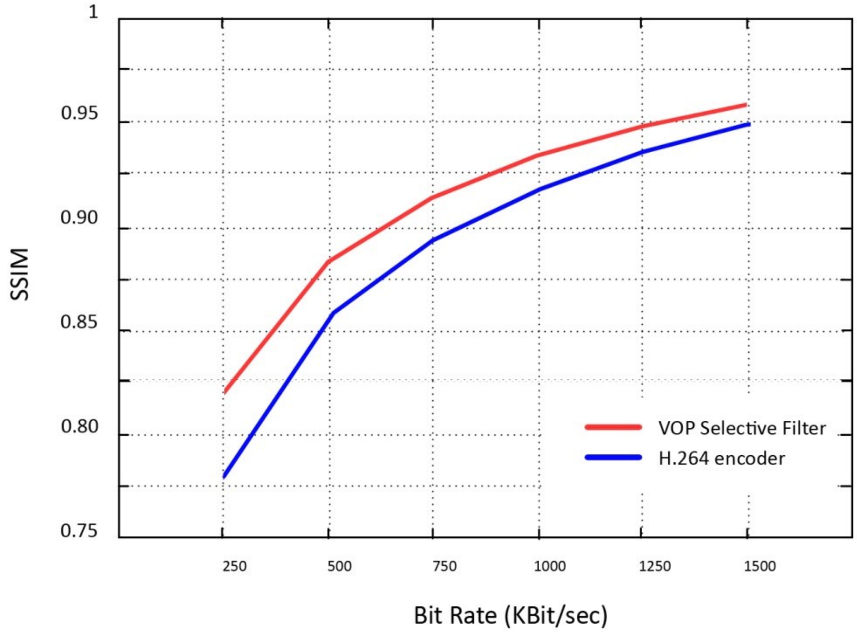


Figure 25: SSIM for overall frame for goldfish 1080p (a) Our approach with VOP Selective filter (b) H.264 encoder without any VOP. The binary mask on (a) is 25%.

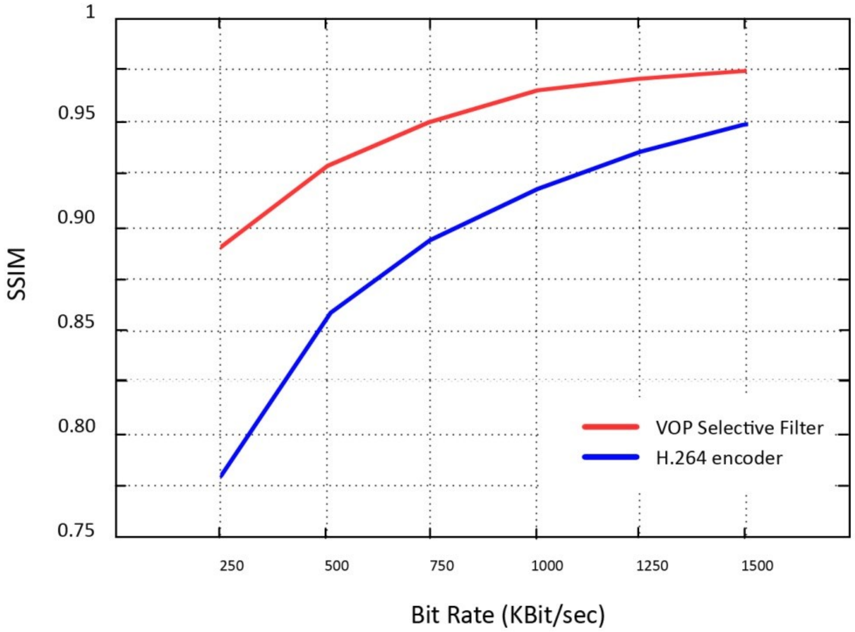


Figure 26: SSIM for masked area for goldfish 1080p (a) Masked area for VOP selective filter (b) H.264 encoder. Note that the H.264 encoder has no improvement whether the SSIM is done on whole frame or part of it.

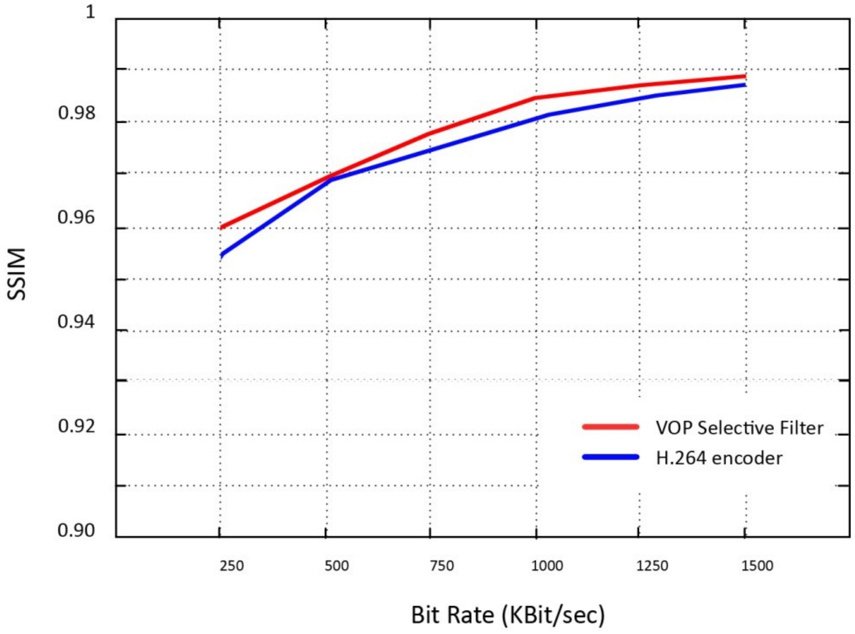


Figure 27: SSIM for overall frame for goldfish 720p (a) Our approach with VOP Selective filter (b) H.264 encoder without any VOP. The binary mask on (a) is 25%.

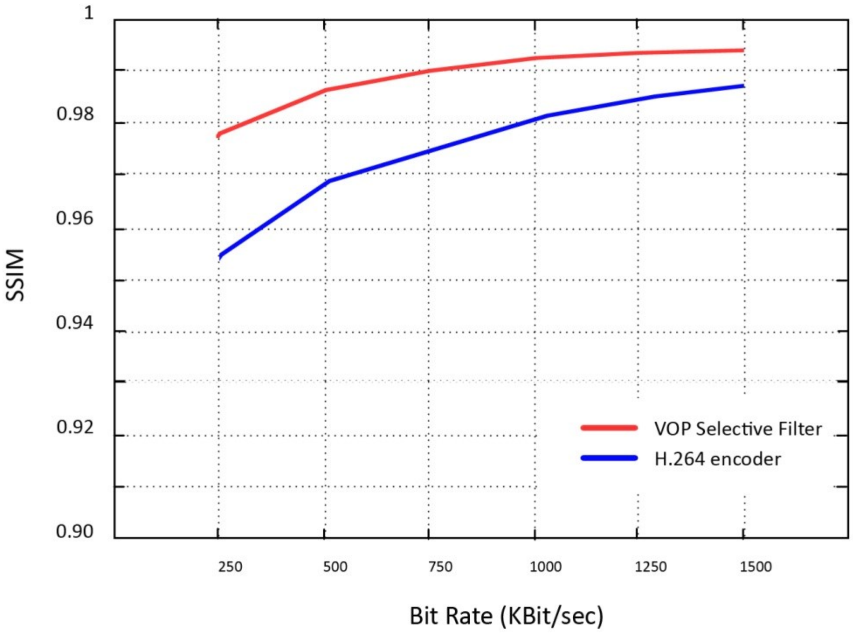


Figure 28: SSIM for masked area for goldfish 720p (a) Masked area for VOP selective filter (b) H.264 encoder.

The SSIM analysis shows that if taken alone (figure 26 and 27), the VOP has SSIM performance which is more than 20% better.

6.5 Case 2: Water Fall

A video of waterfall was shot to analyze the selective filter. The video is a shot of waterfall but in the middle of the video, the camera zooms in at the base thus increasing the size of the binary mask. First 42 frames of the video are shown in figure 29:

	Waterfall 1080p	Waterfall 720p
Height of frame (px)	1080	720
Width of frame (px)	1920	1280

	Waterfall 1080p	Waterfall 720p
Frame rate (fps)	29.97	29.97
Total size (in MB)	156	110
Total running time (in sec)	30	31

Table 5: Parameters for Waterfall 1080p and 720p

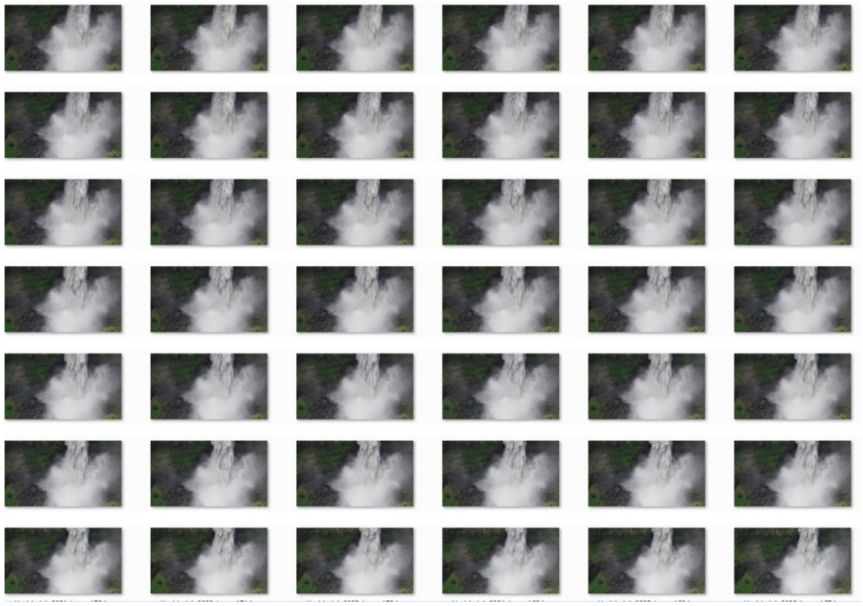


Figure 29: Waterfall video showing first 42 frames of the sequence. The video is shot at 1080p and then downsized to 720p.

The waterfall video is a special case where the binary mask is adopted based on the change in the area of interest. Figure 30 shows the binary mask used in 15 seconds of the video while figure 31 shows the mask used after camera zooms in. The bit rate of this video is shown in figure 33. As it can be seen at 15 second mark, the video bit rate increases and is more than that of the H.264 encoder.

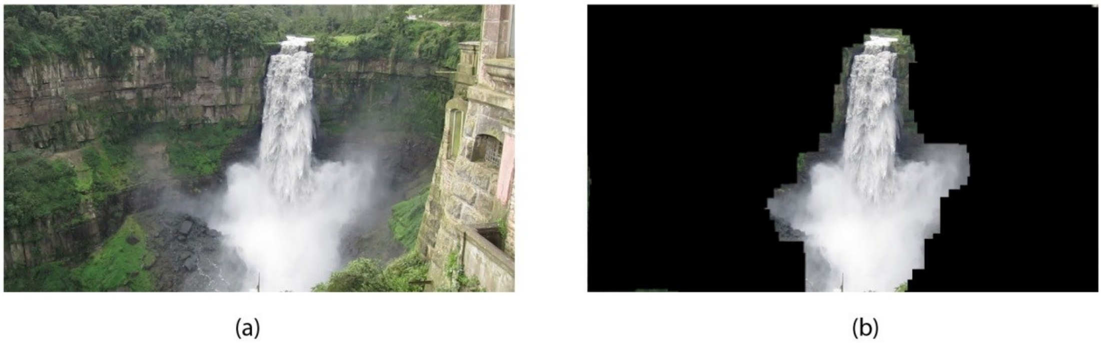


Figure 30: Binary mask for initial 15 seconds of the video while the camera is still.

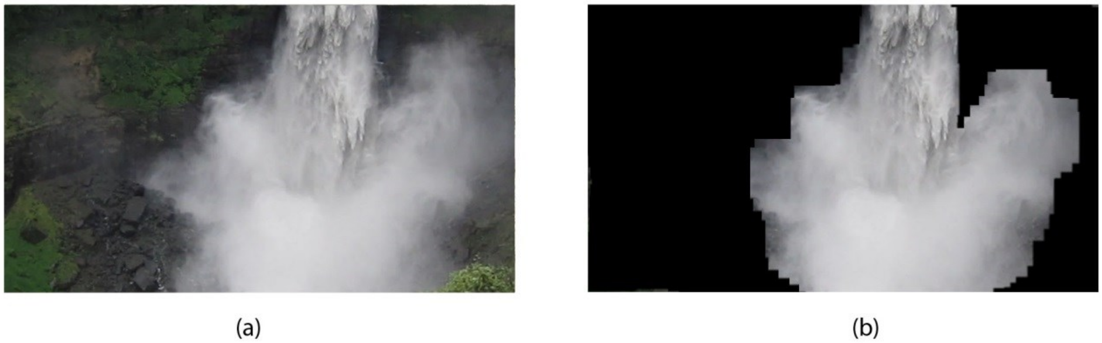


Figure 31: Binary mask for the last 15 seconds of the video once the camera has zoomed in.

As discussed before, the VOP's binary mask adjusts itself to the size and shape of object being encoded. In figure 30b, the mask covers the complete waterfall which is approximately 25% of the frame in this case. At 15 second mark, when the camera zooms in on the waterfall, the binary mask changes its size to 55% (figure 31b). This change in mask is achieved by analyzing the motion vector between two consecutive frames. If these vectors are positive, the mask increases in size, if it is negative the mask shrinks to cover to desired VOP. In figure 33 it is evident that bit rate increase rapidly at 15 second mark. This is the frame where the camera zooms in and VOP increases. The bit rate in this case is more than that of an H.264 encoder.

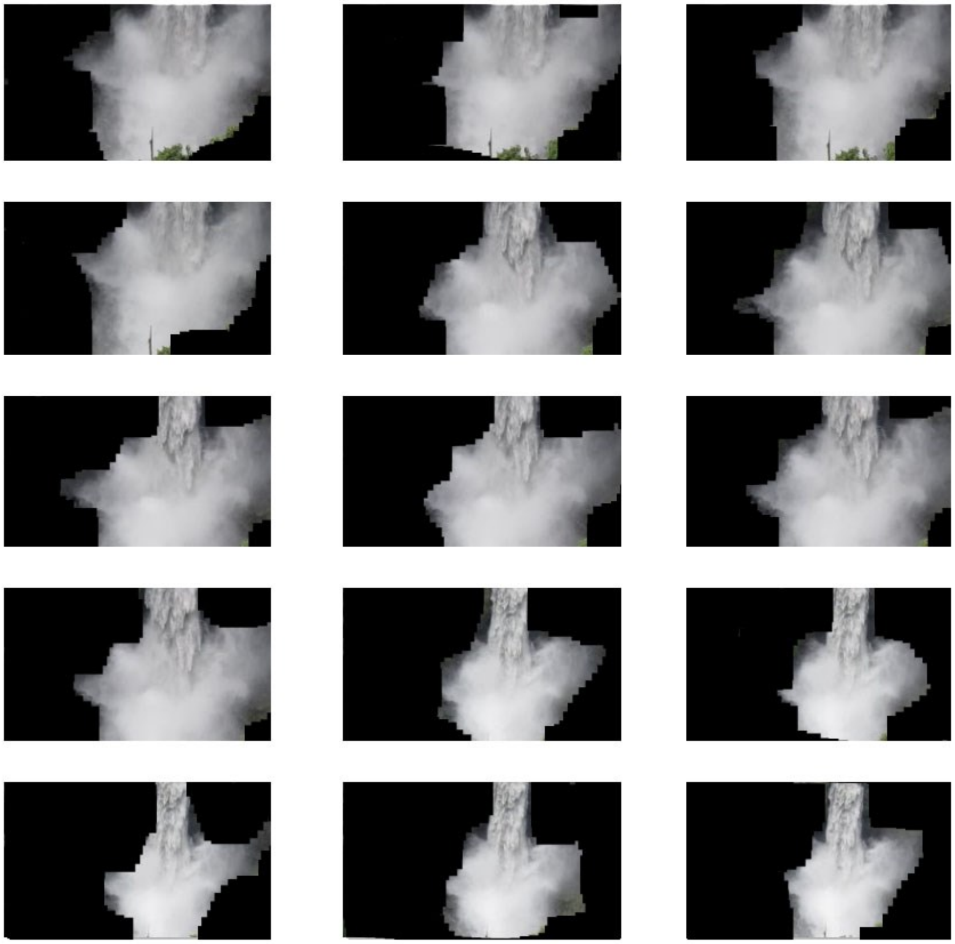


Figure 32: Masking on initial frames of waterfall video

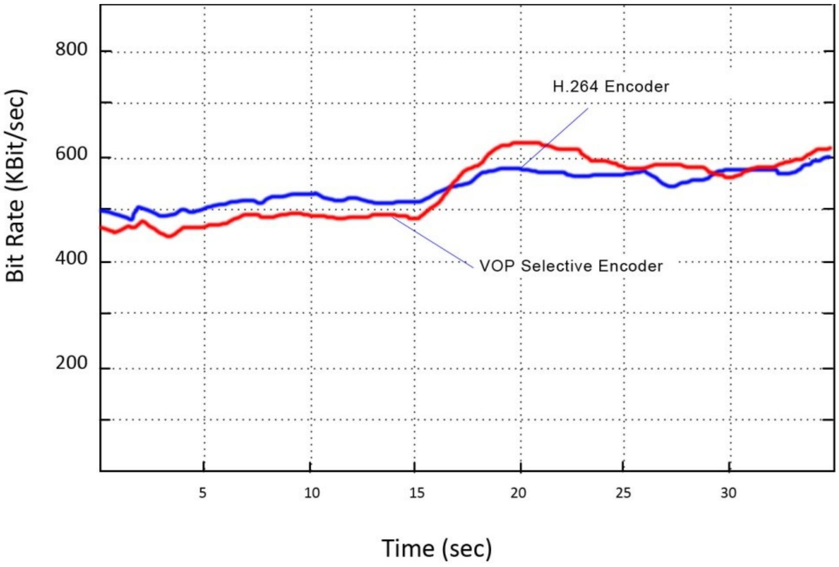


Figure 33: Bit rate of waterfall 720p for VOP selective filter and H.264 encoder for same quality and same PSNR

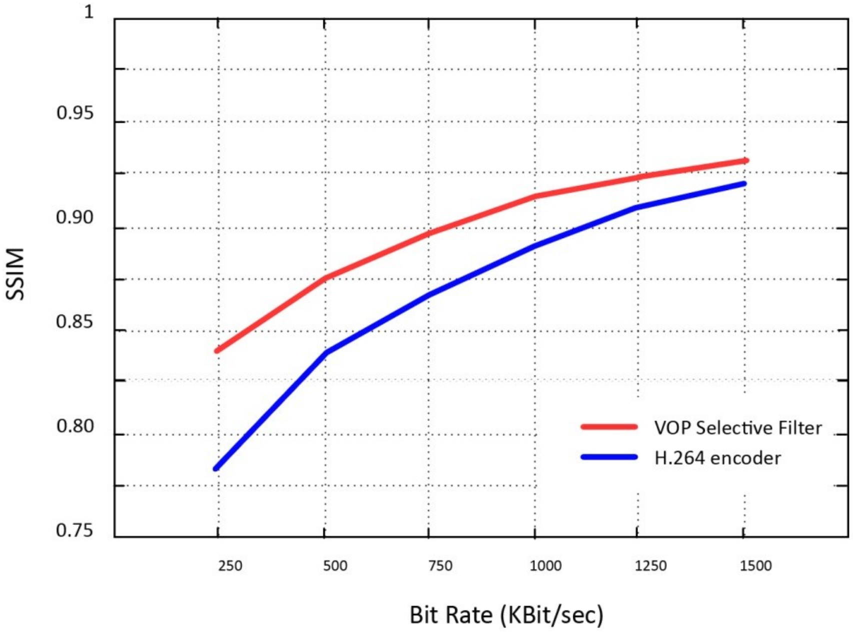


Figure 34: SSIM for full frame for Waterfall 1080p (a) Masked area for VOP selective filter (b) H.264 encoder without any VOP.

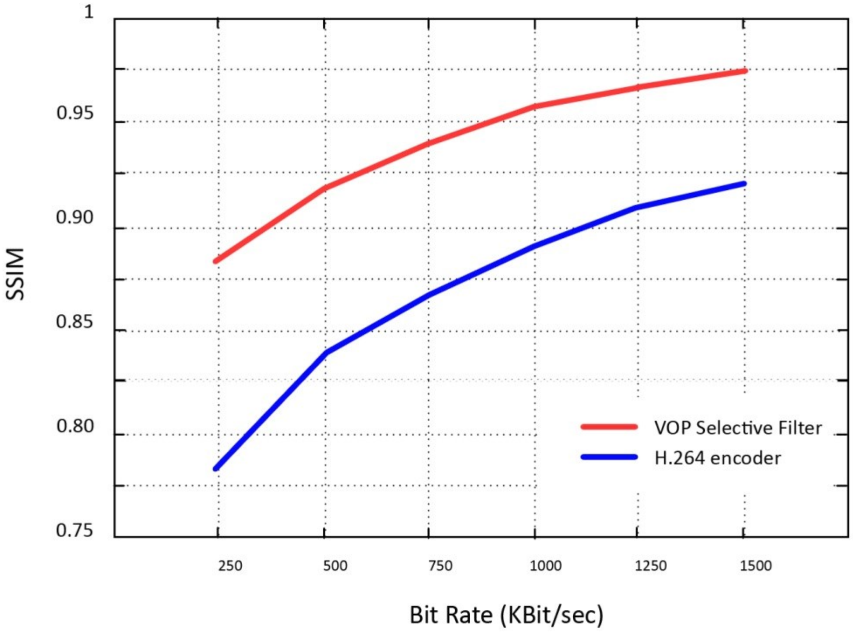


Figure 35: SSIM for masked area for waterfall 1080p (a) Masked area for VOP selective filter (b) H.264 encoder. Note that the H.264 encoder has no improvement whether the SSIM is done on whole frame or part of it.

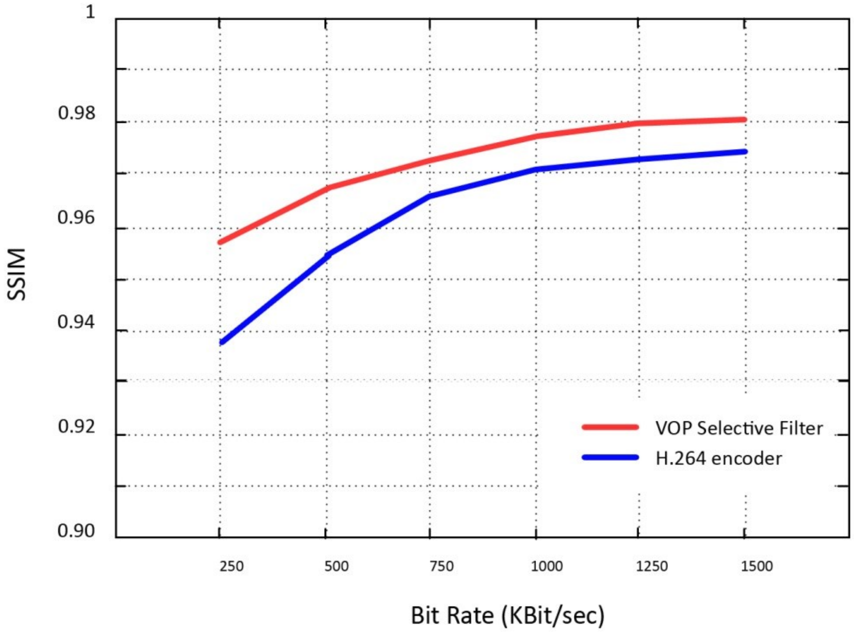


Figure 36: SSIM for full frame for waterfall 720p (a) VOP selective filter (b) H.264 encoder without any VOP.

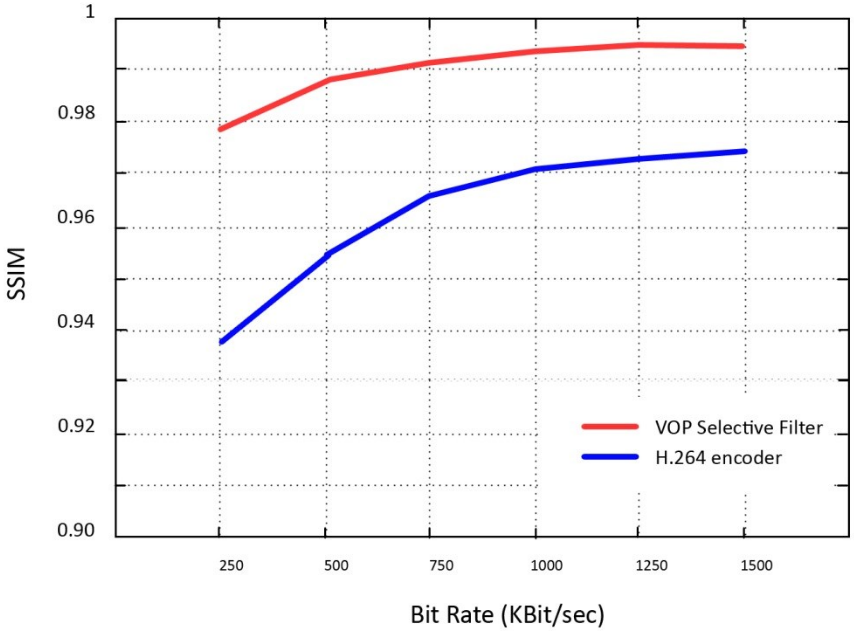


Figure 37: SSIM for masked area for waterfall 720p (a) Masked area for VOP selective filter (b) H.264 encoder. Note that the H.264 encoder has no improvement whether the SSIM is done on whole frame or part of it.

As described above, the area of interest in initial frames in waterfall video is roughly 25% but it increases to 55% after camera zooms in. Figure 38 shows the SSIM of the last 15 seconds of the waterfall 1080p video when the frame is 55% area of interest the SSIM performance is below for this case and VOP loses its edge. In figure 39 the unmasked area is considered and it can be seen that the performance drops even further. This drop in performance causes the SSIM to fall even more than that of H.264.

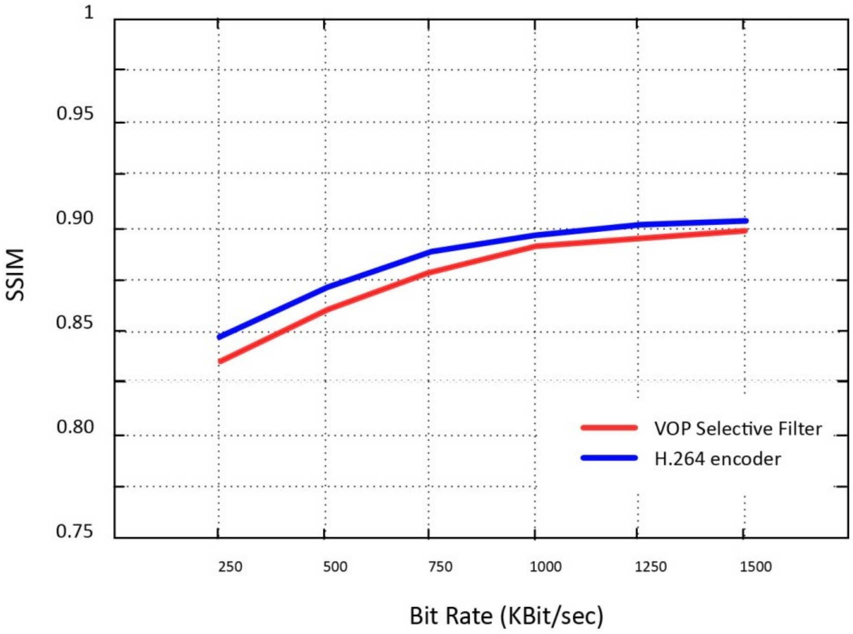


Figure 38: SSIM for full frame of waterfall 1080p when the water fall covers 55% of the frame (a) VOP selective filter (b) H.264 encoder without any VOP. The binary mask on (a) is 55%.

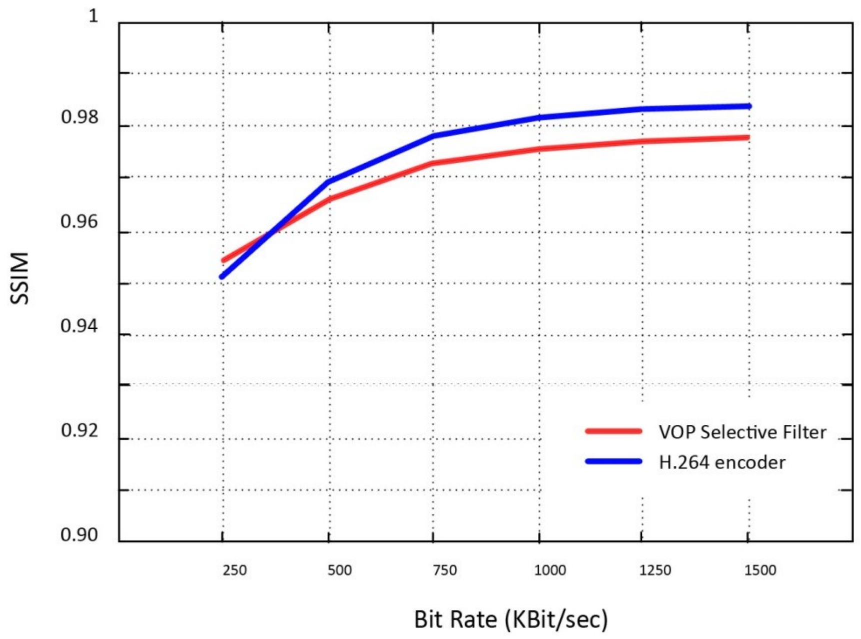


Figure 39: SSIM for full frame of waterfall 720p when the water fall covers 55% of the frame (a) VOP selective filter (b) H.264 encoder without any VOP. The binary mask on (a) is 55%.

From the result of SSIM analysis on frame which cover the 55% of frame, it is evident that selective filter suffers in performance and falls below the H.264 encoder without selective filter.

6.6 Case 3: Peacock

Third video was shot of a peacock. The blue head and chest of the peacock were selected for binary mask. The peacock video contained an element which moves at faster rate and has irregular shape. The tracking of binary mask is shown in figure 41.

This video consists of 25% selective binary mask on an object which moves fast. The fine tuning of mask is shown in figure 42

Height of frame (px)	720
Width of frame (px)	1280
Frame rate (fps)	29.97
Total size (in MB)	54
Total running time (in sec)	10

Table 6: Parameters for Peacock 720p

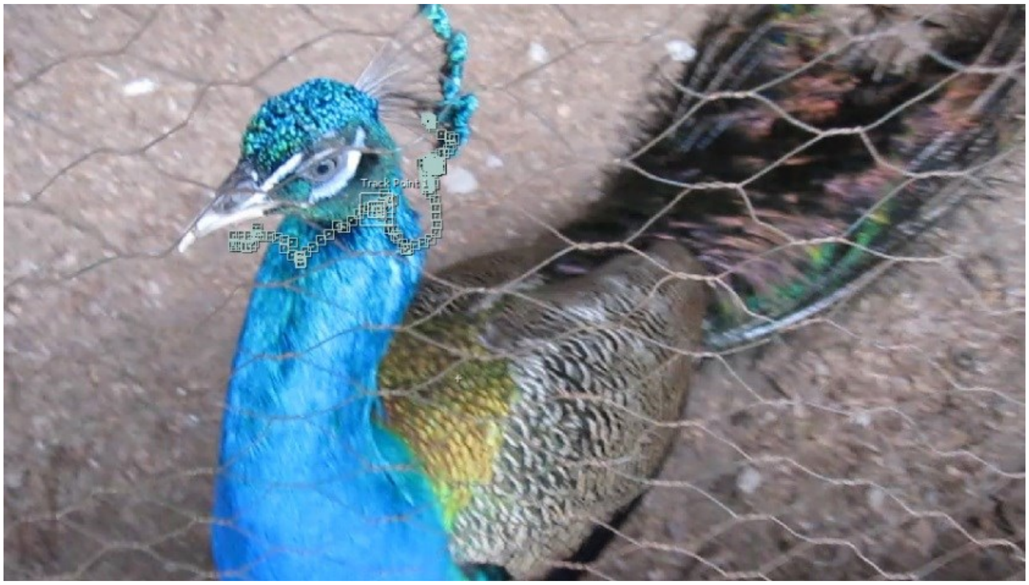


Figure 40: Binary mask tracking of peacock head

Figure 40 shows the tracking of binary mask around the peacock's head. It is interesting to note that binary mask moves back and forth at a fast rate. This fast movement of head is difficult to track and due to the fact that this movement is translational as well as rotational.

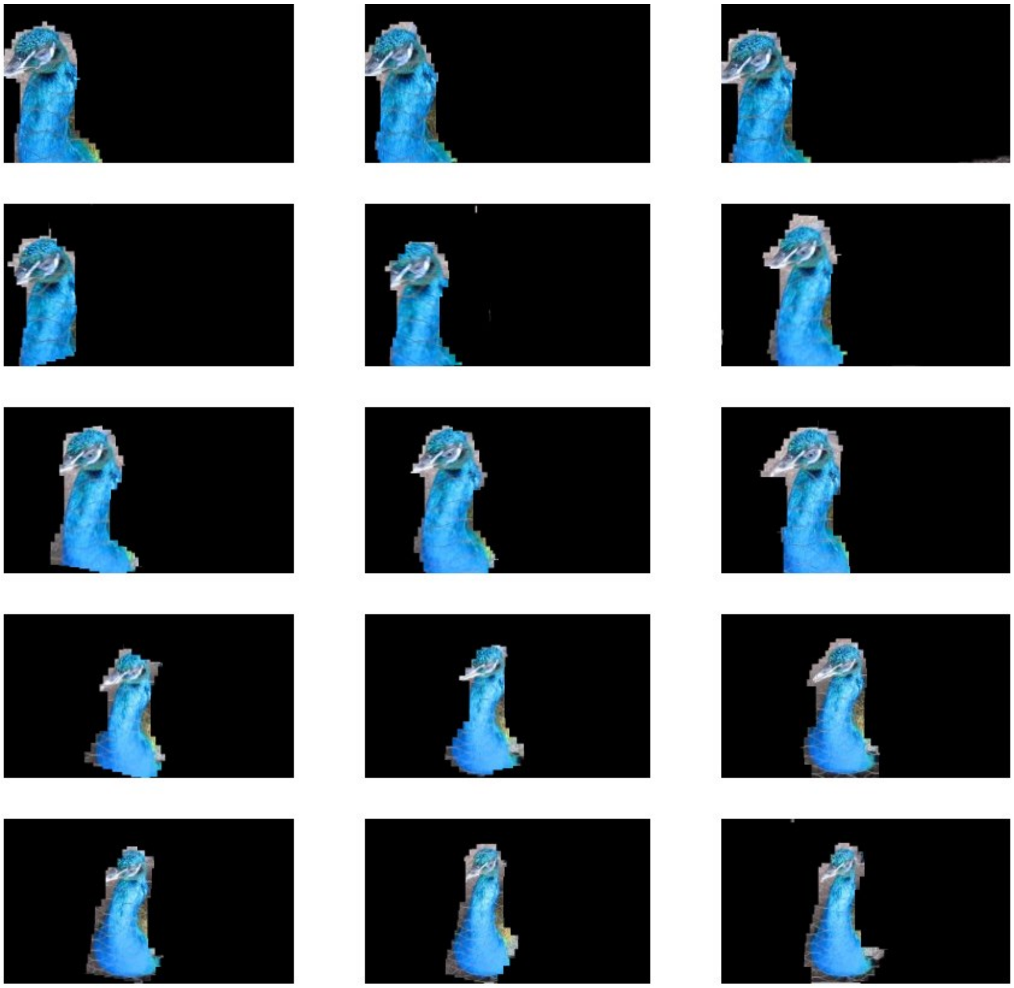


Figure 41: Binary mask fine tuning of a fast moving object (in this case a peacock)

The binary mask tracking for peacock head is challenging because the head moves at fast pace and the contrast ratios of the frame are not good. But the results are better compared to H.264 encoder. The reason for better results is the fact that fur of a peacock is repetitive pattern and more intra-frame prediction can be used to encode VOP.

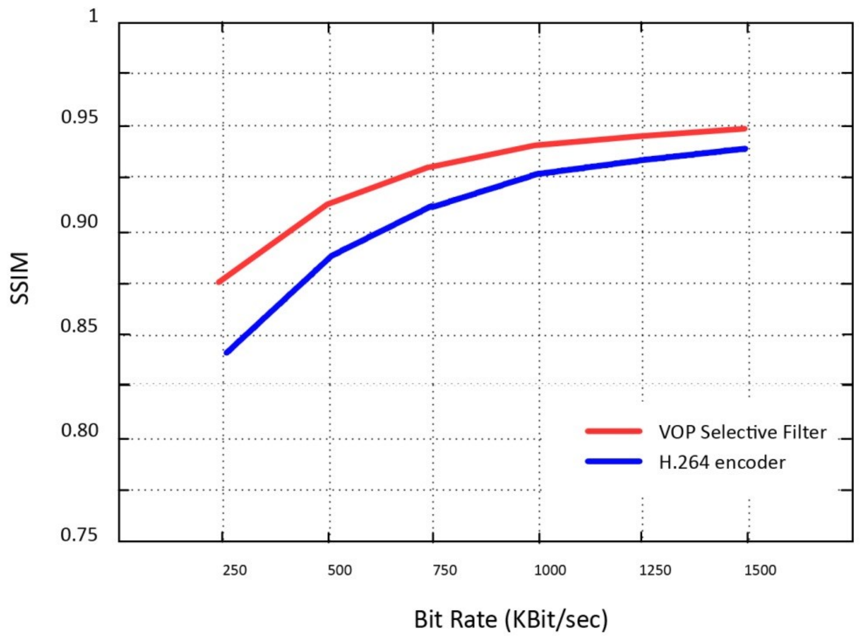


Figure 42: SSIM for masked area for peacock 720p (a) VOP selective filter (b) H.264 encoder without filter

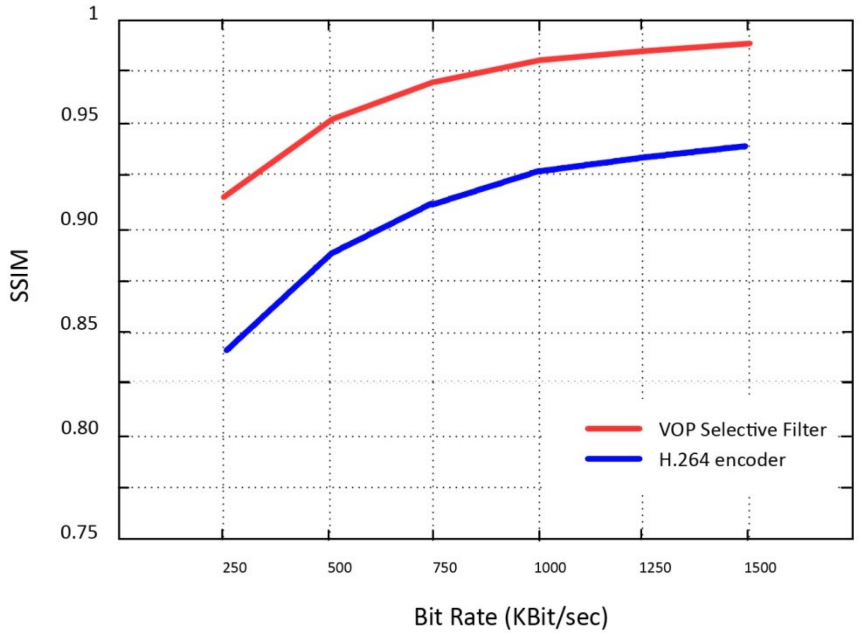


Figure 43: SSIM for masked area for peacock 720p (a) Masked area for VOP selective filter (b) H.264 encoder. Note that the H.264 encoder has no improvement whether the SSIM is done on whole frame or part of it.

From the above SSIM analysis we can see that masked area has better SSIM compared to an H.264 encoder without any selective filter. The masked area is more like an uncompressed frame in the encoded video.

6.7 MSE and PSNR

Mean squared error as the name specifies is average of square of errors and provides the level of error between compressed and uncompressed frame. Mathematically, it is defined as:

$$MSE(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{13}$$

Where the N is number of pixels, x_i and y_i are the *i*th samples in image x and y.

PSNR (peak signal to noise ratio) is one of the most commonly used criteria for comparing image or video compressions [26]. It is related to MSE by following formula:

$$PSNR = 10 \log_{10}(\frac{M^2}{MSE}) \tag{14}$$

Where *M* is the maximum possible intensity of pixels in the frame.

The algorithm achieves 43 dB PSNR at 1 Mbps (figure 44). This is comparable to other H.264/AVC encoders [38] and significantly better than previous H.263 standard codecs [39] for same resolution.

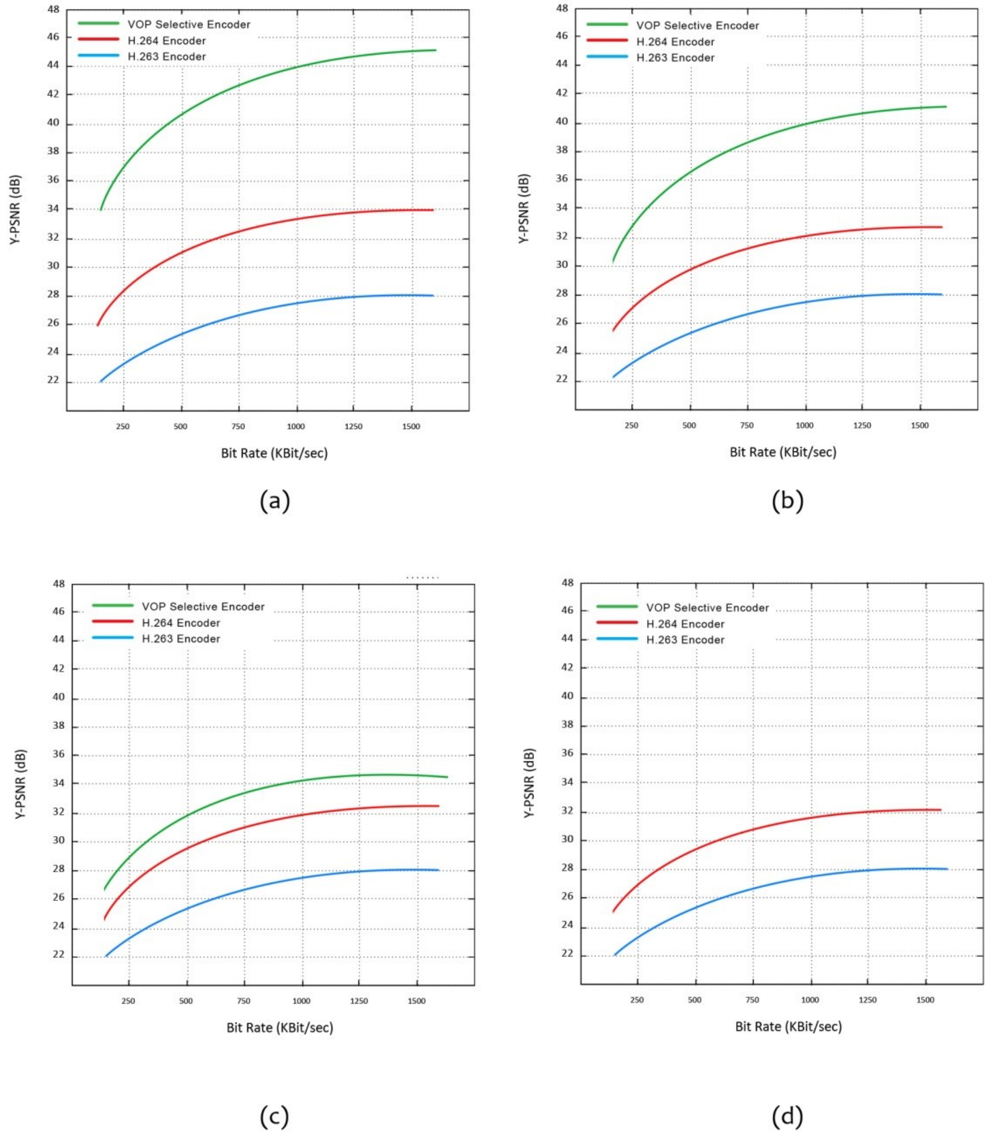
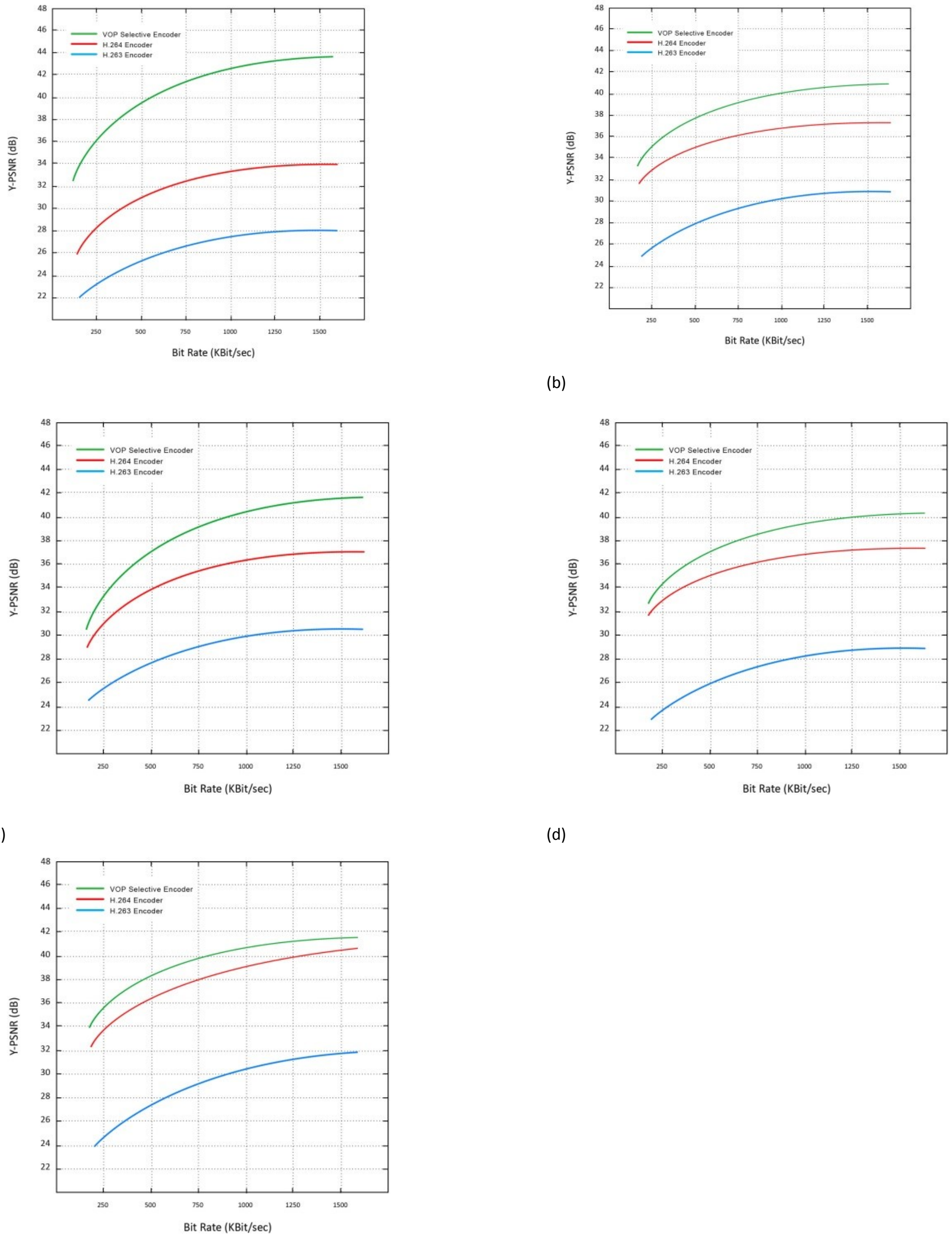


Figure 44: Comparison of three encoders for Goldfish 1080p. The VOP was selected progressively to increase the area of interest (H.264 with selective filter, H.264 without filter and H.263) with different percentage of area of interest chosen. (a) 20% (b) 35% (c)



(b)

(c)

(d)

Figure 45:PSNR comparison of five videos encoded using three different encoders. (a) Goldfish 1080p (b) Goldfish 720p (c) Waterfall 1080p (d) Waterfall 720p (e) Peacock 720p

PSNR analysis on the all of five videos show better performance compared to standard encoders. This performance is dependent on type of subject and area of video object plane.

6.8 Power Analysis

With increasing use of computer being mobile nowadays, it is necessary to analyze the power consumption of our scheme. The power consumption analysis proposed by[40] has been used to analyze the power used by processor while encoding the video.

$$P_{h264} = P_{encoder} + P_{selective\,filter} \tag{15}$$

$$P_{h264} = f_{tran}^{h264}(F_l + F_B, W, H) + f_{mask}(F_s, W, H) \tag{16}$$

Where F_l and F_B are the frame rates of I and B frames, W is the width, H is the height and F_s is the binary mask pixel coverage area. As in [40], both functions (f_{tran}^{h264} and f_{mask}) depend on number of pixels processed per unit time. For the purpose of power analysis we used an Intel Core i3 processor clocked at 2.2 GHz equipped with 4GB of RAM and running Windows 7 operating system. In figure 46, it is evident that VOP consumes less power than a standard Intel IPP and JM based encoders at lower bit rates but at higher bit rates it consumes more power due to that fact that it is processing higher pixel number for same bit rate compared to other encoders.

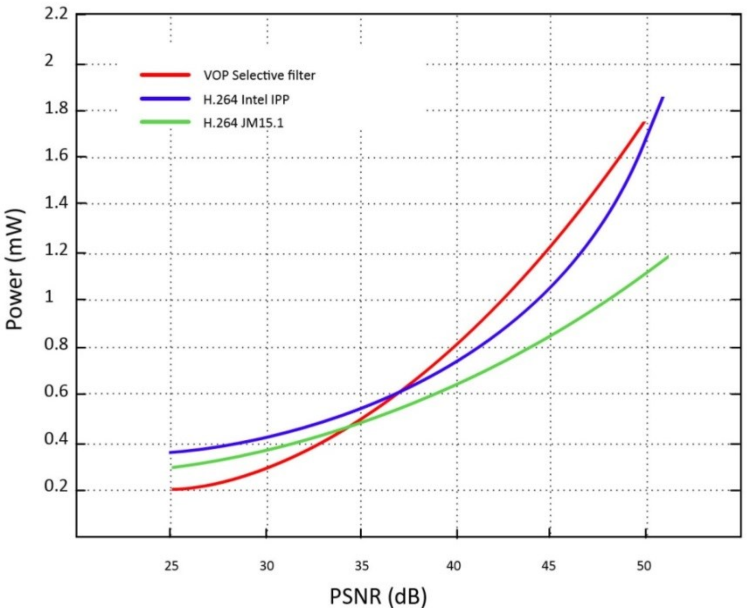


Figure 46: Power consumption analysis of three encoders (H.264 with selective filter, H.264 Intel IPP without filter and H.264 JM15.1) with different percentage of area of interest chosen.

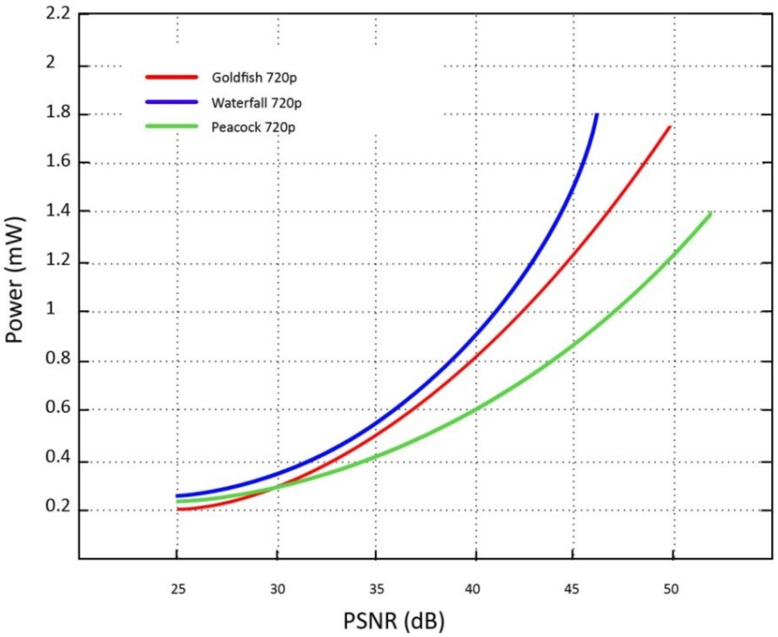


Figure 47: Power consumption analysis of VOP for three videos.

6.9 Compression Analysis

It is important to analyze how much compression we have achieved with our selective filter technique over common H.264 encoder. For this purpose we recorded the compressed videos and looked at the size of videos. The results for our selective filter, H.264 encoder and H.263 encoder are shown below.

Bit rate (Kbits/Sec)	VOP	H.264	H.263
500	88	86	46
750	78	62	38
1000	62	49	32
1250	56	43	27
1500	48	36	22

Table 7: Compression ratios of Godfish 1080p

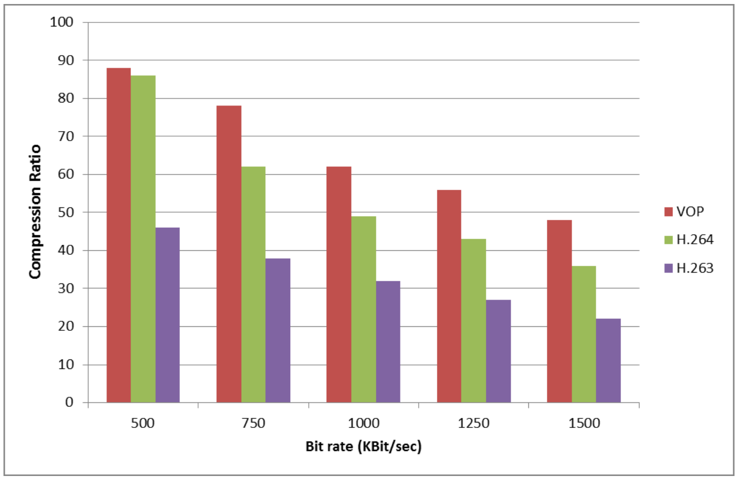


Figure 48: Compression ratios of Godfish 1080p

Bit rate (Kbits/Sec)	VOP	H.264	H.263
500	88	88	51
750	75	74	48
1000	65	53	36
1250	56	43	30

Bit rate (Kbits/Sec)	VOP	H.264	H.263
1500	45	38	24

Table 8: Compression ratios of Godfish 720p

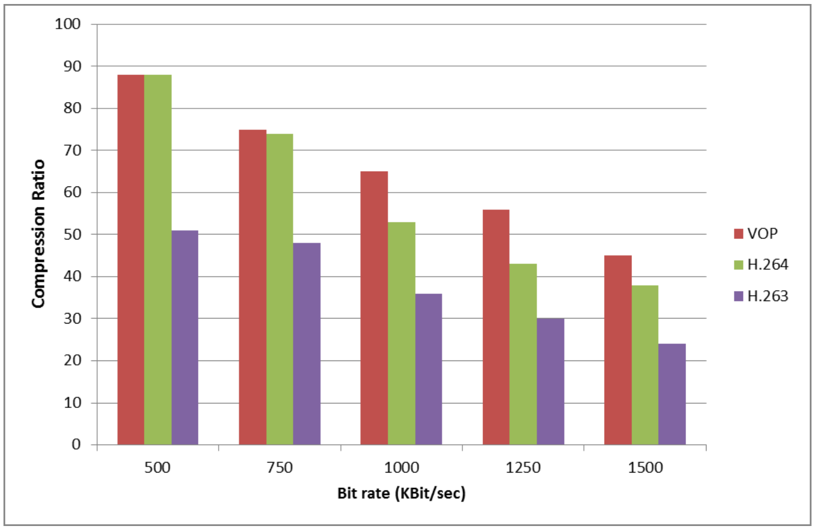


Figure 49: Compression ratios of Godfish 720p

For waterfall videos, low compression is achieved due to big VOP area. This increased area of VOP causes the compression ratio to decrease significantly and at 750 Kbit/sec, the standard encoder performs at exactly the same ratio (figure 50).

Bit rate (Kbits/Sec)	VOP	H.264	H.263
500	82	86	34
750	75	75	28
1000	63	53	24
1250	56	45	20
1500	47	38	16

Table 9: Compression ratios of Waterfall 1080p

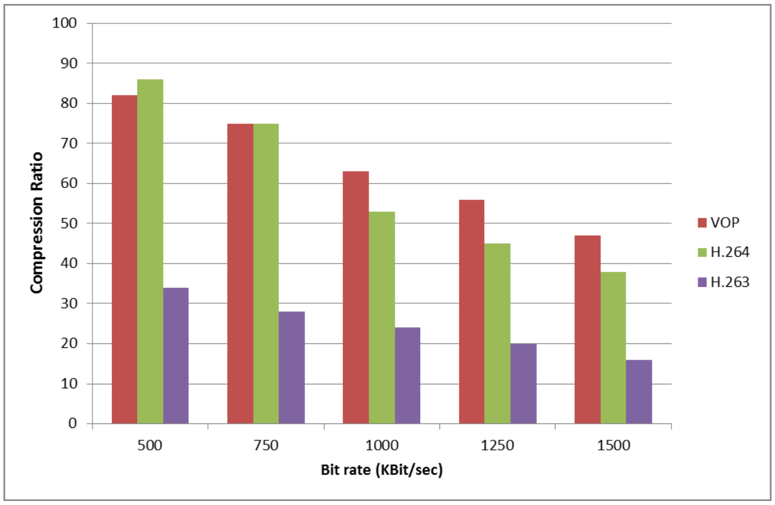


Figure 50: Compression ratios of Waterfall 1080p

Bit rate (Kbits/Sec)	VOP	H.264	H.263
500	89	84	36
750	80	76	31
1000	71	59	27
1250	64	47	24
1500	52	41	19

Table 10: Compression ratios of Waterfall 720p

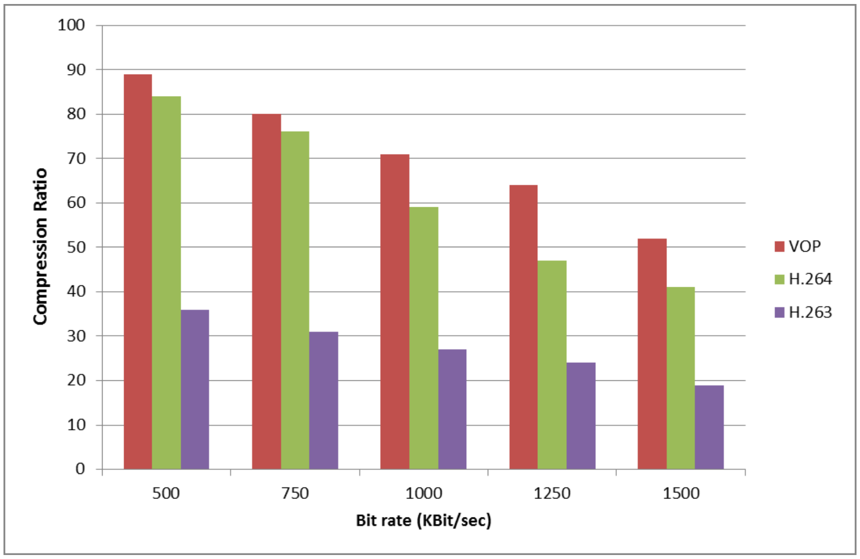


Figure 51: : Compression ratios of Waterfall 720p

Bit rate (Kbits/Sec)	VOP	H.264	H.263
500	76	81	25
750	72	73	20
1000	64	59	17
1250	59	51	15
1500	52	48	15

Table 11: Compression ratios of Peacock 720p

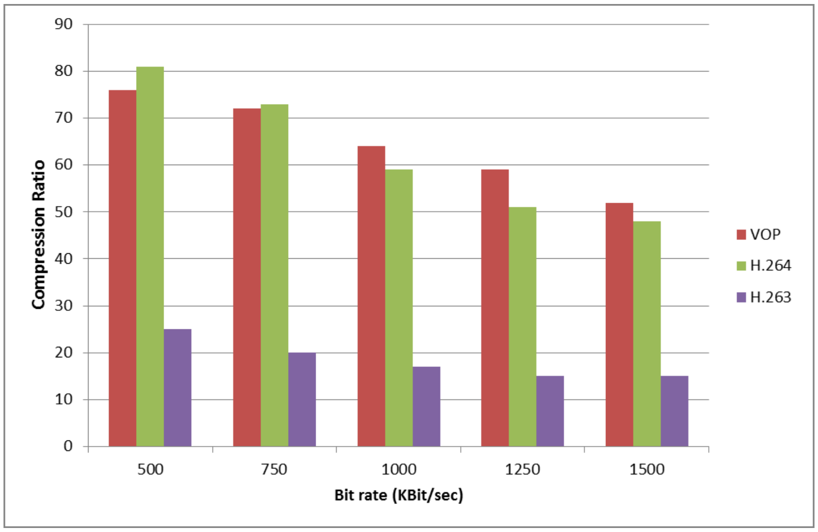


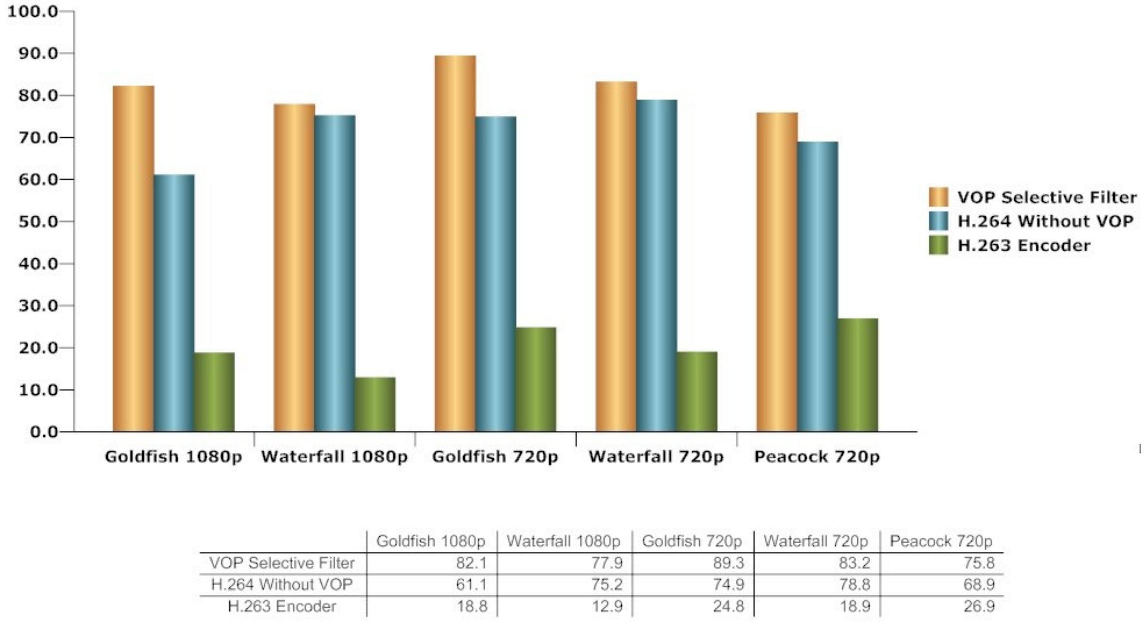
Figure 52: Compression ratios of Peacock 720p

6.10 Human Perception Survey

Although the SSIM and PSNR are good ways of measuring performance of an encoder, we also conducted human perception survey to see how human perceive the different encoded videos. For this purpose we distributed all five videos (shortened to 20 second clips) encoded with different encoders and ask survey participants to give us their opinion. The questions asked were:

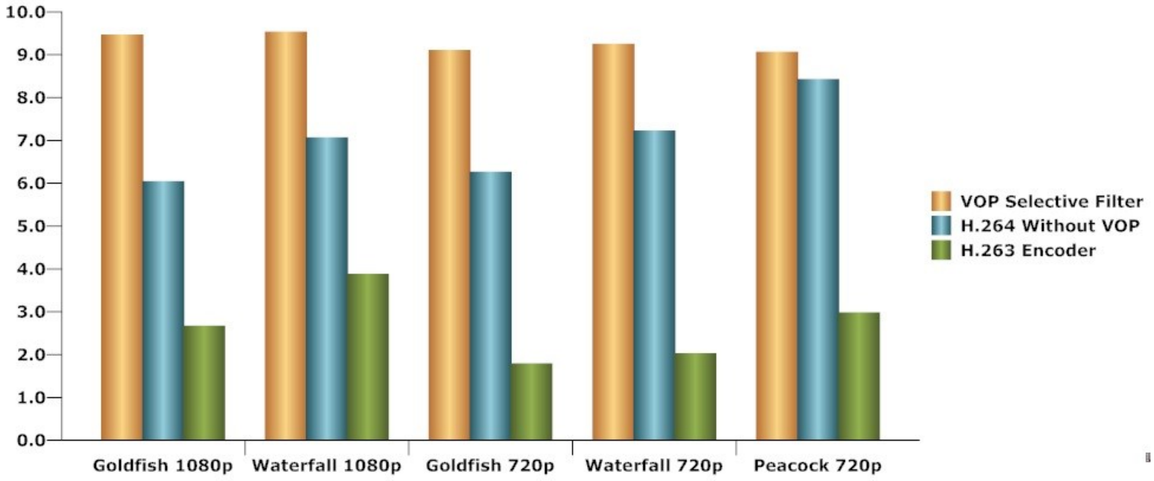
- Rate the overall video quality 1 to 10 (with 10 being best)
- Rate the contrast of videos 1 to 10 (10 being best)
- Rate the pixelation of video on 1 to 10 (10 being best)
- Rate the noise in video on 1 to 10 (10 being best)

We used Survey Money (*www.surveymonkey.com*) for this purpose. After one week we got 300 replies to our survey. Out of sample set of 300, 100% of them had watched online video streaming in the past. 73% had a broadband internet connection and screen resolution of 1024x768 or above. This sample set of 219 respondents with similar equipment and internet connectivity was chosen for data analysis. The results of this survey (figure 53 to 56) were analogous to what we predicted from our PSNR analysis.



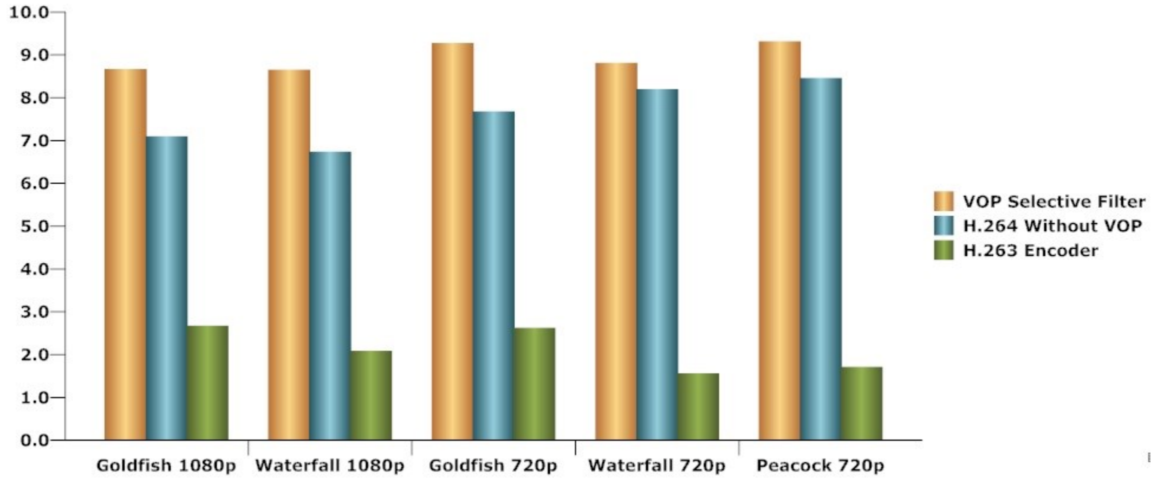
	Goldfish 1080p	Waterfall 1080p	Goldfish 720p	Waterfall 720p	Peacock 720p
VOP Selective Filter	82.1	77.9	89.3	83.2	75.8
H.264 Without VOP	61.1	75.2	74.9	78.8	68.9
H.263 Encoder	18.8	12.9	24.8	18.9	26.9

Figure 53: Overall quality comparison based on human perception survey



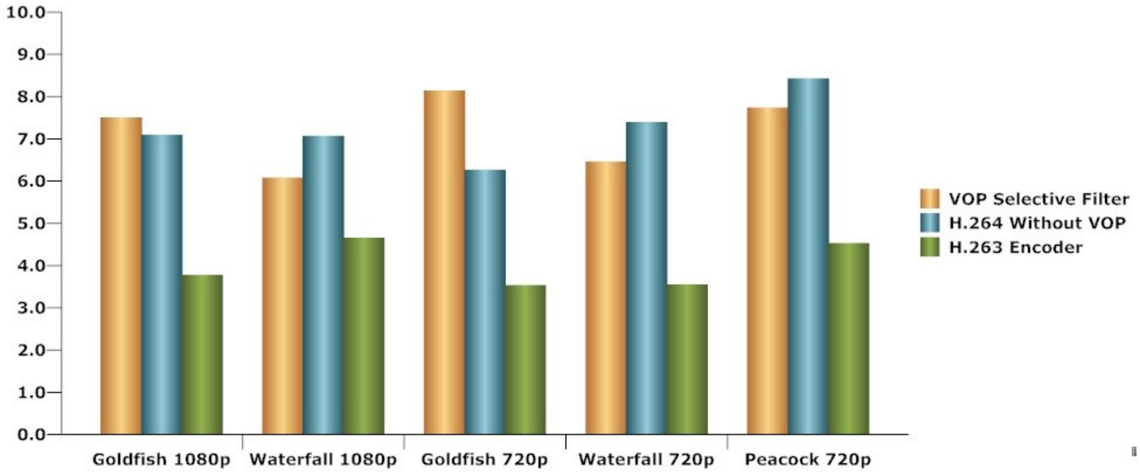
	Goldfish 1080p	Waterfall 1080p	Goldfish 720p	Waterfall 720p	Peacock 720p
VOP Selective Filter	9.5	9.5	9.1	9.2	9.1
H.264 Without VOP	6.0	7.1	6.3	7.2	8.4
H.263 Encoder	2.7	3.9	1.8	2.0	3.0

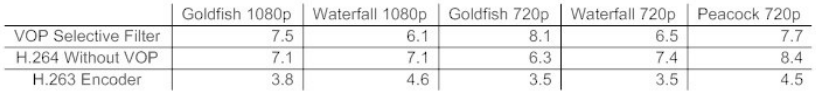
Figure 54: Average score for contrast comparison based on human perception survey



	Goldfish 1080p	Waterfall 1080p	Goldfish 720p	Waterfall 720p	Peacock 720p
VOP Selective Filter	8.7	8.6	9.3	8.8	9.3
H.264 Without VOP	7.1	6.7	7.7	8.2	8.5
H.263 Encoder	2.7	2.1	2.6	1.5	1.7

Figure 55: Average score for noise comparison based on human perception survey





	Goldfish 1080p	Waterfall 1080p	Goldfish 720p	Waterfall 720p	Peacock 720p
VOP Selective Filter	7.5	6.1	8.1	6.5	7.7
H.264 Without VOP	7.1	7.1	6.3	7.4	8.4
H.263 Encoder	3.8	4.6	3.5	3.5	4.5

Figure 56: Average score for pixelation in video based on human perception survey

It can be seen from the above graphs (figures 53-56) that our encoder performed better for all videos in all cases whether it was contrast or low noise. The only limitation was seen in waterfall video. The reason for this is the increased VOP once the camera zooms in on the scene. This increase VOP requires higher bit rates and processing power. In cases where the bit rate is limited, the pixelation becomes evident. These results are consistent with SSIM and PSNR analysis of our VOP encoder. The results of our VOP have been published in [41].

Chapter 7: Conclusions and Future Works

7.1 Conclusion

In this thesis we developed a new approach for video encoding where the user has more control over what needs to be encoded at high frame rate and at high resolution. This approach can be used in cases where the motion of object being filmed is relatively slow and the high definition video is desired. The results of our video selective filter encoder has better performance compared to other encoders and use of multi-core processor makes the task of encoding the video in real time possible. Our approach of developing video selective filter separate from encoder makes it a flexible system to use. Any video encoder (As far as it complies with ITU specifications) can be used with our selective filter. This approach makes it an excellent system to use in application where hardware based encoders are being used. By running our selective filter on a separate chip and then using the output as an input to encoder to encode on another chip of the concerned processor can give better results without investing huge amount of money and time into developing a new encoder. The use of commonly available game controller is also an interesting approach as it makes it less expensive to deploy this system. The results of our research are summarised below:

- Our VOP approach has better performance (SSIM, PSNR and HPS) compared to H.264 encoder for videos with less than 30% VOP area. Above this threshold the performance drops and in cases when VOP is above 55%, the VOP method is more computationally intensive than standard H.264 and has lower PSNR and SSIM.

- At same PSNR and with 30% VOP, our approach has lower bit rate. This makes it useful to transmit high definition video on slower internet connections.

- VOP has lower power consumption (processing of pixel per unit time) at low PSNR values. Making it a good alternative for mobile chips (in current form, it only works in Intel based architecture.)

- VOP is encoded at high pixel density for selected area compared to standard encoders. Thus even at low bit rate, the VOP is clear with low pixelation.

- High contrast and compression ratio results have been verified using human perception survey. These results show that VOP has better contrast ratio than H.264.

- One of the limitations of our encoder is that it only works for slow moving object. VOP gives better results when the object vibration or movement is below 22 movements or vibrations per second. A heart beat is considered slow in this case.

- Pixelation issues with large VOP are observed due to the fact that computational needs increase as the VOP increases in size. This limitation can be overcome by using powerful processors and high bit rates.

7.2 Future Works

Any future work should focus on developing hardware based selective filter. We reckon that such a system will be fast and will be able to give better results even when the objects are fast moving. Another recommendation is to write this selective filter code for AMD processors. A development for ARM architecture is possible but the presently available processors are not powerful enough to handle the computations needed to solve the binary mask.

Another future development can be use of boundary detection to further make the binary mask accurate and store it in memory. This way the scene change issue can be solved and once the object moves back into the frame, the software can start tracking the object again.

Bibliography

[1] J. J. Craig and M. Raibert. “A systematic method of hybrid position/force control of a manipulator.” *The IEEE Computer Society's Third International Computer Software and Applications Conference,* 1979.

[2] H. Asada, J.J.E. Slotine. “Robot Analysis and Control.” *JWS Press,* 1986.

[3] Z. Lu, S. Kawamura and A. A. Goldenberg. “Sliding mode impedance control and its application to grinding tasks.” *International Workshop on Intelligent Robots and Systems,* 1991.

[4] N. Hogan. “Impedance control: An approach to manipulation.” *American Control Conference,* 1984.

[5] Jong Hyeon Park. “Impedance control for biped robot locomotion.” *IEEE Transactions on Robotics and Automation,* 17(6), pp. 870-882.

[6] D. W. Marhefka and D. E. Orin. “Simulation of contact using a nonlinear damping model.” *IEEE International Conference on Robotics and Automation,* 1996.

[7] R. Johansson and M. W. Spong. “Quadratic optimization of impedance control.” *IEEE International Conference on Robotics and Automation,* 1994.

[8] M. Matinfar and K. Hashtrudi-Zaad. “Optimization-based robot impedance controller design.” *IEEE Conference on Decision and Control,* 2004.

[9] D. Bertsekas. “Dynamic Programming and Optimal Control.” *Athena Scientific*.

[10] R. E. Bellman. “Dynamic Programming.” *Princeton Univ. Press*.

[11] R. M. e. a. Murray. “A Mathematical Introduction to Robotic Manipulator.” *CRC Press*., 1994.

[12] J. P. Quirion, E. Gunn and J. Gu. “Optimal control of permanent magnet motors using dynamic programming.” *IEEE Conference on Robotics, Automation and Mechatronics,* 2004.

[13] C. Seong. "Neural dynamic programming and its application." *Stanford University*.

[14] O. Khatib. “A unified approach for motion and force control of robot manipulators: The operational space formulation.” *IEEE Journal of Robotics and Automation,* 3(1), pp. 43-53.

[15] R. Anderson and M. W. Spong. “Hybrid impedance control of robotic manipulators.” *IEEE International Conference on Robotics and Automation,* 1987

[16] D. E. Whitney. “Historical perspective and state of the art in robot force control.” *IEEE International Conference on Robotics and Automation,* 1985.

[17] L. M. Hocking. “Optimal Control: An Introduction to the Theory with Application.” *Oxford University Press*.

[18] G. Ferretti, G. A. Magnani and P. Rocco. “Impedance control for elastic joints industrial manipulators.” *IEEE Transactions on Robotics and Automation*, 20(3), pp. 488-498.

[19] Hun-ok Lim, S. A. Setiawan and A. Takanishi. “Balance and impedance control for biped humanoid robot locomotion.” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.

[20] R. Ikeura, T. Moriguchi and K. Mizutani. “Optimal variable impedance control for a robot and its application to lifting an object with a human.” *11th IEEE International Workshop on Robot and Human Interactive Communication*, 2002.

[21] S. X. Yang, M. Meng and Xiaobu Yuan. “A biological inspired neural network approach to real-time collision-free motion planning of a nonholonomic car-like robot.” *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2000.

[22] D. Kirk.”Optimal Control Theory.” *Prentice-Hall.*

[23] Microsoft, "XNA developer center.” *http://msdn.microsoft.com/en-us/centrum-xna.aspx* (4/19/2013).

[24] D. P. Noonan, G. P. Mylonas, A. Darzi and Guang-Zhong Yang. “Gaze contingent articulated robot control for robot assisted minimally invasive surgery.” *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2008.

[25] ITU. "H.264 : Advanced video coding for generic audiovisual services " *http://www.itu.int/rec/T-REC-H.264* (4/19/2013).

[26] K. V. S. Swaroop and K. R. Rao. “Performance analysis and comparison of JM 15.1 and intel IPP H.264 encoder and decoder.” *42nd Southeastern Symposium on System Theory (SSST),* 2010.

[27] A. Jazayeri and M. Tavakoli. “A passivity criterion for sampled-data bilateral teleoperation systems.” *IEEE World Haptics Conference (WHC),* 2011.

[28] Qiang Liu, R. J. Sclabassi, M. L. Scheuer and Mingui Sun. “A two-step method for compression of medical monitoring video.” *The 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2003.

[29] Munchurl Kim, Jae Gark Choi, Daehee Kim, Hyung Lee, Myoung-Ho Lee, C. Ahn and Yo-Sung Ho. “A VOP generation tool: Automatic segmentation of moving objects in image sequences based on spatio-temporal information.” *IEEE Transactions on Circuits and Systems for Video Technology,* 9(8), pp. 1216-1226.

[30] A. R. Graves and C. Czarnecki. “Distributed generic control for multiple types of telerobot.” *IEEE International Conference on Robotics and Automation,* 1999.

[31] P. Velrajkumar, S. S. Manohar, A. Cv, A. D. J. Raju and R. Arshad. “Development of real-time tracking and control mobile robot using video capturing feature for unmanned applications.” *IEEE International Conference on Communication Control and Computing Technologies (ICCCCT),* 2010.

[32] T. Kawai, T. Fukuda, M. Nako, Y. Yasuda, E. Murata, E. Kaku and K. Tatsuno. “Remote visitor robot through the internet.” *International Symposium on Micro-Nano Mechatronics and Human Science,* 2009.

[33] B. Trammell and D. Schatzmann. “A tale of two outages: A study of the skype network in distress.” *7th International on Wireless Communications and Mobile Computing Conference (IWCMC),* 2011.

[34] N. Khezami, S. Otmane and M. Mallem. “An approach to modelling collaborative teleoperation.” *International Conference on Advanced Robotics,* 2005.

[35] Adobe Systems, "Statistics: PC penetration." *http://www.adobe.com/products/flashplatformruntimes/statistics.html* (2/13/2013).

[36] S. K. Chatterjee and I. Chakrabarti. “A high performance VLSI architecture for fast two-step search algorithm for sub-pixel motion estimation.” IMPACT '09. International Multimedia, Signal Processing and Communication Technologies, 2009.

[37] Hongtao Yu, Zhiping Lin and Feng Pan. “Applications and improvement of H.264 in medical video compression.” *IEEE Transactions on Circuits and Systems I: Regular Papers,* 52(12), pp. 2707-2716.

[38] Kun Ouyang, Qing Ouyang and Zhengda Zhou. “Optimization and implementation of H.264 encoder on symmetric multi-processor platform.” *WRI World Congress on. Computer Science and Information Engineering,* 2009.

[39] Lin Tong and K. R. Rao. “Region of interest based H.263 compatible codec and its rate control for low bit rate video conferencing.” *International Symposium on Intelligent Signal Processing and Communication Systems,* 2005.

[40] A. Ukhanova, E. Belyaev and S. Forchhammer. “Encoder power consumption comparison of distributed video codec and H.264/AVC in low-complexity mode.” *International Conference on Software, Telecommunications and Computer Networks (SoftCOM),* 2010.

[41] M. Khan and J. Gu. “Web based teleoperation architecture and H.264 video encoder.” *IEEE Canadian Conference on Electrical & Computer Engineering (CCECE),* 2012.