An Industry Oriented Mini Project Report on

A CONVEX OPTIMIZATION FRAMEWORK FOR VIDEO QUALITY AND RESOLUTION ENHANCEMENT FROM MULTIPLE DESCRIPTIONS

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

PEETHANI NAGASREE	**198R1A0401**
AMBALA NARESH	**198R1A0402**
ANAND TERATI	**198R1A0403**
ANKITA	**198R1A0404**

Under the Guidance of

Dr. BHASKER DAPPURI

Professor

DEPARTMENT OF ECE

ENGINEERING COLLEGE

EXPLORE TO INVENT

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(Approved by AICTE, Affiliated to JNTUH, Accredited by NBA, NAAC)

Kandlakoya (v), Medchal, Telangana.

2022-23

CMR ENGINEERING COLLEGE

(Approved by AICTE, Affiliated to JNTUH, Accredited by NBA, NAAC)

Kandlakoya (v), Medchal, Telangana.

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the Industry Oriented Mini-Project work entitled **"A CONVEX OPTIMIZATION FRAMEWORK FOR VIDEO QUALITY AND RESOLUTION ENHANCEMENT FROM MULTIPLE DESCRIPTIONS"** is being submitted by **Dr. BHASKER DAPPURI** bearing Roll No:**198R1A0401**, **PEETHANI NAGASREE** bearing Roll No:**198R1A0402, AMBALA NARESH** bearing Roll No:**198R1A0403, ANAND TERATI** bearing Roll No:**198R1A0404, ANKITA** in Batch IV-1 semester, Electronics and Communication Engineering is a record bonafide work carried out by them during the academic year 2022-2023.The results embodied in this report have not been submitted to any other University for the award of any degree.

INTERNAL GUIDE | HEAD OF THE DEPARTMENT

Dr. BHASKER DAPPURI | **Dr. SUMAN MISHRA**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We sincerely thank the management of our college **CMR ENGINEERING COLLEGE** for providing during our project work.

We derive great pleasure in expressing our sincere gratitude to our principal **Dr. A.S.REDDY** for his timely suggestions, which helped us to complete the project work successfully.

It is the very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA**, Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Dr. T. SATYANARAYANA,** Associate professor, Department of ECE for his ideas that led to complete the project work and we also thank his continuous guidance, support and unfailing patience, throughout the course of this work.

We sincerely thank our project internal guide **Dr. BHASKER DAPPURI,** Professor, Department of ECE for his guidance and encouragement in carrying out this project.

DECLARATION

We hereby declare that the project work entitled **"A CONVEX OPTIMIZATION FRAMEWORK FOR VIDEO QUALITY AND RESOLUTION ENHANCEMENT FROM MULTIPLE DESCRIPTIONS"** is the work done by us in campus at **CMR ENGINEERING COLLEGE,** Kandlakoya during the academic year 2022-23 and is submitted as Industry oriented Mini Project in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD.**

PEETHANI NAGASREE	**198R1A0401**
AMBALA NARESH	**198R1A0402**
ANAND TERATI	**198R1A0403**
ANKITA	**198R1A0404**

CONTENTS

LIST OF FIGURES

ABSTRACT

Transmission and compression technologies advancement over the past decade led to a shift of multimedia content towards cloud systems. Multiple copies of the same video are available through numerous distribution systems. Different compression levels, algorithms and resolutions are used to match the requirements of particular applications.

As 4k display technologies are rapidly adopted, resolution enhancement algorithms are of vital importance. Current solutions do not take into account the particularities of different video encoders, while video reconstruction methods from compressed sources do not provide resolution enhancement.

In this project, we propose a multi-source compressed video enhancement framework where each description can have a different compression level and resolution. Using a variation formulation based on a modern proximal dual splitting algorithm, we efficiently combine multiple descriptions of the same video.

Super-resolution (SR) algorithms are post-processing techniques that infer a spatially High Resolution (HR) estimate from one or more Low Resolution (LR) images.

Two applications are proposed: combining two compressed Low Resolution descriptions of a video sequence into a High Resolution description and enhancing a compressed HR video using a LR compressed description.

Tests are performed over multiple video sequences encoded with High Efficiency Video Coding (HEVC), at different compression levels and resolutions obtained through multiple down-sampling methods.

CHAPTER 1

1.1 INTRODUCTION

The continuous evolution of transmission systems, storage and video compression technology in the past decade provides the end user with easy access to video content. A varied number of distribution methods exist, from the classical DVD's to online streaming on the World Wide Web. Large video databases such as YouTube or Netflix provide multiple resolutions and different encodings of the same video in order to account for user bandwidths and displays. This situation creates a lot of potential for resolution enhancement and compression artifact reduction techniques from single and multiple sources. The ability to efficiently combine multiple descriptions of the same video and exploit the information variety can be a useful tool in several scenarios. Video transmission systems that rely on scalable encoders, such as Scalable High Efficiency Video Coding (SHVC), generate multiple representations of a video sequence at a different quality level or resolution. The information variety between these representations can be used to obtain a higher quality video. Similar scenarios can be encountered when enhancing older videos that are only available in compressed form or when working with MultiView plus Depth (MVD) video transmission systems where the problem of image alignment is achieved through depth computed disparity. Super-resolution (SR) algorithms are post-processing techniques that infer a spatially High Resolution estimate from one or more Low Resolution images. In general, SR algorithms can be divided in single-frame (SF-SR) or multi-frame (MF-SR) approaches. The later exploits the motion between successive LR frames in order to extract unique information from each representation. In the case of 3D video, adjacent views can also be used. The problem formulation in most cases assumes the availability of a high number of descriptions (5-30) which are subjected to different pixel shift operations (rotations, translations), blurring and sub-sampling. This type of MF-SR approaches is best suited to tackle the problem of image acquisition, where a high number of descriptions is available with simple motion and a similar blurring. In the recent work of Liu and Sun. In this scenario, the descriptions are consecutive frames of a video sequence. As noted by the authors, this problem is inherently more difficult as real world videos have complex motion rather than a simple parametric form. The paper proposes practical SR frameworks where optical flow, blur kernel and noise levels are simultaneously estimated. Gains of up to 3 dBs are reported over bicubic up-sampling, when super-resolving using 15 forward and backward frames. The degradation was synthetically added as Gaussian blurring, sub-sampling and

Gaussian white noise; tests were performed on real world video sequences. However, as reported by the authors it can take 2 hours to super-resolve one frame. A sequential approach is used for motion estimation using optical flow between current and past frames while the frame deblurring and blur estimation is performed using an iterative multi-scale approach with a Huber-norm based cost function minimized using the conjugate gradient method. They show that compression artifacts complicate the SR reconstruction and suggest that a model of compression should be employed. The method is shown to provide improvements over spatial domain methods on frequency quantized images, when exact motion information is known. The authors propose a practical framework that enhances the quality of video by combining different encodings of the same sequence. In this scenario real world videos are encoded using MPEG2 in two configurations. The proposed algorithm is able to combine the two decoded videos. Gains of up to 1.5 dB are reported, however, no resolution enhancement is performed. Optical flow is used to obtain correspondences between up-sampled LR patches in consecutive frames and the dictionary is trained using all corresponding LR patches and the HR patch. The proposed method shows gains of up to 4 dB over bicubic interpolation, albeit the algorithm was trained on the same sequence and 14 hour were needed for 3 frames input training and up to 1 minute to super resolve due to the motion compensation (motion information was computed before training with optical flow). Nowadays, videos are mainly available in compressed form. Furthermore, different observations of a video sequence are usually available with a different compression level and resolution. To the best of our knowledge, there are no algorithms tackling the problem of super-resolution and compressed video enhancement from multiple descriptions of the same video. More precisely: 1) we extend our model to take into account multiple videos, 2) we model the down sampling process to account for any polyphase filter in a manner consistent with our choice of convex optimization method, 3) we integrate temporal prediction in our model 4) we integrate HEVC compression model and finally 5) we implement the framework using a modern and efficient proximal dual-splitting algorithm.

CHAPTER 2

LITERATURE SURVEY

The general multi-layer high-level syntax design common to all multi-layer HEVC extensions, including SHVC, MV-HEVC, and 3D-HEVC, is described. The inter-layer reference picture processing modules, including texture and motion resampling and color mapping, are also described. Even for those applications that are well-suited to scalable coding, adoption in products has been limited. A key impediment to deployment of past scalable coding standards has been the difficulty of implementation, and the significant implementation differences between scalable and non-scalable coding. In particular, an attempt was made to minimize implementation complexity by enabling repurposing of multiple single-layer HEVC cores to achieve efficient scalable coding. As will be discussed in further detail in Section III, this goal is achieved by adopting a scalable coding architecture that relies on making high-level syntax only changes to the underlying single-layer HEVC standard. As such, SHVC became the first scalable video coding standard that is built upon the HLS-only scalable coding framework. Empowered by efficient inter-layer reference picture processing modules discussed in Section V, SHVC achieves high scalable coding efficiency without requiring any block level coding logic changes to the single-layer HEVC cores. Besides ease of implementation, there are many other design benefits to taking this approach, as will be described in detail later in this paper. With the finalization of the SHVC extensions in 2014, there is hope that this time SHVC will be widely deployed for those applications that can benefit from scalability. Throughout this paper, SHVC is compared with SVC. The multi-view extension of H.264/AVC is called Multiview Video Coding (MVC). The first version of HEVC achieved roughly 50% bit rate reduction over its predecessor AVC at comparable subjective quality. The second version of HEVC includes scalability extensions (SHVC), multi-view extensions (MVHEVC), and format range extensions An overview of HEVC can be found in and of its extensions in. The extension for 3D coding with depth (3D-HEVC) was recently finalized, and work is underway for additional extensions for screen content coding (SCC). Before the first version of HEVC was completed, contributions were submitted to JCT-VC showing evidence of the feasibility of scalable extensions. Plans for future scalable extensions were considered during the design of the high-level syntax of the first version of HEVC. Several features were planned in the first version of HEVC such that the changes required to support future scalable extensions (and other extensions) could be minimized. A typical format for some of these applications is the

Multi View Video (MVV) composed of a set of N video sequences representing the same scene, referred to as views, acquired simultaneously by a system of N cameras positioned under different spatial configurations. An alternative representation is the Multiview-Video-Plus-Depth format (MVD) [1], where the depth information is used in addition to texture for each viewpoint. This allows for a less costly synthesis of much more virtual views, using for example Depth-Image-Based Rendering (DIBR) methods. View synthesis is the process of extrapolating or interpolating a view from other available views. It is a popular research topic in computer vision, and numerous methods have been developed in this field over the past four decades. View synthesis techniques can be mainly classified in three categories [2]. The methods in the first category, like DIBR, require explicit geometry information such as depth or disparity maps to warp the pixels in the available views to the correct position in the synthesized view. Methods in the second category require only implicit geometry, like some pixel correspondences in the available and synthesized view that can be computed using optical flow for instance. Finally, methods in the third category require no geometry at all. These areas appear as holes in virtual views, also referred to as disocclusions. This problem is currently resolved by using in painting algorithms such as the ones described in [3] and [4]. Recently, the Moving Pictures Experts Group (MPEG) expressed a significant interest in MVD formats for their ability to support 3D video applications. This new activity is mainly focused on developing a 3D extension of the HEVC [5] video coding standard, after a first standardization activity finalized with Multiview Video Coding (MVC) [6]. An experimental framework was developed as well, in order to conduct the evaluation experiments. Due to a big temporal distance between reference and synthesized frames, the motion vector fields may be imprecise especially for frames with intense motion. We mitigate these effects by using the so-called “Hierarchical” synthesis scheme, in which temporal layers are used to perform symmetric synthesis (where each frame is synthesized from either a key frame or a previously synthesized frame) and we compare it with a “Direct” scheme (where each frame is directly synthesized from a past and a future key frame). To further improve the quality of the synthesis, we introduce an adaptive fusion method that selects between inter-view and temporal prediction. The remaining disocclusions in the synthesized image are then filled by a linear in painting method. In addition, a standard quantitative method for assessing the relative merit of super-resolution algorithms is required. Focuses on multi-frame super-resolution techniques, which aim to improve image resolution by fusing information from multiple low resolution images. Multi-frame super-resolution algorithms require several slightly different perspectives of the same scene, a non-integer pixel shift between images, such that each input view captures a marginally different representation of the scene. The subtle differences between images are then exploited to create an image that exceeds the spatial resolution of the input frames. While a wide array of super-resolution algorithms now exist, far less attention has been paid to evaluating and comparing the capability of these techniques in practical scenarios. When publishing a new algorithm a majority of researchers provide results obtained using their method but seldom compare its standing relative to existing techniques. In addition, super-resolution algorithms are most commonly evaluated subjectively, further inhibiting comparison. Hence, the primary goal of this paper is to present a comprehensive and practical comparison of existing super-resolution techniques. In total, 13 different super-resolution algorithms are evaluated using a common platform. Each method is examined using 4 common grayscale images; algorithm execution times are also compared along with visual results. When assessing the practical capabilities of different approaches to super resolution it is imperative that image registration is also investigated, as techniques rely on the smallest of differences between frames. While registration parameters are often assumed known, registration algorithms used in conjunction with super-resolution are also included in our analysis. This paper also aims to identify the image quality metric most suited to the appraisal of super-resolved images. In the event that super-resolution results are objectively assessed this is often limited to evaluating the peak signal-to-noise ratio (PSNR), with respect to an original high-resolution image, compared to bicubic or cubic-spline interpolation. While subjective inspections are indeed an ideal evaluation method, in practice the time and resources needed to conduct a full subjective assessment of each super resolved image produced are simply not available. In practical situations, a method of quickly measuring the quality of a super-resolved image is required. However, this should not imply that other metrics cannot accurately assess super resolved image quality. The universal quality index (UQI) and pixel-based visual information fidelity (VIFP) are noticeably absent from existing super-resolution performance evaluations. In order to ensure that super-resolution results are assessed using a broad range of image quality measures, 12 different metrics have been used in this review. We hope that by providing the datasets used and ground truth registration data online this will encourage other researchers to repeat our experiments with different datasets or to evaluate their own algorithms and report their comparative results, leading to benchmark image quality results for new methods in the field. We evaluate some state-of-the-art SR algorithms and the results suggest that different SR algorithms should be utilized for different applications. In general, they reflect the consistency and conflict between objective and subjective measures as well as computer vision systems and human vision systems do. Super-resolution, often called “SR image reconstruction”, is a signal processing approach to overcoming the resolution limitations of low-resolution images. Many SR methods have been proposed in recent years and a comprehensive overview of them is given. Super-resolution methods can be broadly classified into three groups: 1) classical multi-image SR such as in 2) example-based SR methods 3) a combination of both. A recent proposal uses only a single image. In the first group, i.e., classical SR, high-frequency information is assumed to be split across multiple LR images, and can be implicitly found in an aliased form. In the second group, i.e., example-based SR, this missing high-resolution information is assumed to be available in HR database patches and learned from LR/HR pairs of examples in the database. The third and last group combines multi-image SR and example-based SR by searching through image patches of a single image with different scales. However, how to evaluate SR results remains an open problem. Full reference, reduced reference and no reference are three commonly used methods for evaluating image quality. Because SR images are generally generated from LR images, full reference is the best way to evaluate SR quality. The use of common methods for measuring the quality of SR images in the full reference method such as Mean Square Error (MSE) or Peak Signal-to-Noise Ratio (PSNR) is problematic because small registration errors often occur in SR reconstruction based on multiple LR images. In many SR works, the results are often simply evaluated by human observers. We also argue that the subjective evaluation by human observers should be quantified. We propose a task-oriented evaluation method for SR techniques in this study. The major contributions of this study are as follows: 1) Our evaluation method, including both objective and subjective measures, can evaluate SR results more systematically than has been possible heretofore. 2) The objective measures are task-oriented and can cope with the registration errors that are common in SR. 3) The subjective measures are also task-oriented and the results are quantified as scores in our study. 4) We clarified the consistency and conflicts that exist between objective measures related to computer vision systems and subjective measures related to human vision systems. This study intends to establish a sound testing and evaluation methodology based upon the human visual characteristics for appreciating the image restoration accuracy; in addition to comparing the subjective results with predictions by some objective evaluation methods. In total, six different super resolution algorithms - such as iterative back-projection (IBP), robust SR, maximum a posteriori (MAP), projections onto convex sets (POCS), a non-uniform interpolation, and frequency domain approach - were selected. The performance comparison between the SR algorithms in terms of their restoration accuracy was carried out through both subjectively and objectively. The term "super" evokes the characteristics of the technique overcoming the inherent resolution limitation of LR imaging systems.1 This technique has been adopted by various practical applications including medical imaging, satellite imaging, and video technology where multiple image capturing can be made for the same scene. Other applications such as surveillance and forensic investigations often require resolution enhancements for creation of synthetic zooming in order to improve the visibility of a region of interest. Recently, there has been an effort to convert broadcasting video sequence to a higher resolution. Numerous efforts have been done to solve the SR reconstruction problem in the literature.1-8 Mostly, they relate to restoration theory and are shown to be an inverse problem, where an unknown image is to be reconstructed based upon measurements related to it through linear operators and additive noise. The linear relation is composed of geometric warp, blur, and decimation operations. There are more intuitive approaches to the SR image reconstruction using non-uniform interpolation6 and frequency domain based methods. A brief review about some of those representative algorithms are given subsequently. While a considerably large number of SR reconstruction algorithms have been proposed, far fewer visual experiments have been conducted to evaluate the performance and merits of the algorithms. The current study aims to conduct a set of psychophysical assessments and to compare the image restoration accuracy of well known, previously published SR reconstruction algorithms. The results can also contribute to provide a guideline for a subjective evaluation of SR reconstruction methods.

2.1 EXISTING SYSTEM:

Super-resolution algorithms are post-processing techniques that infer a spatially High Resolution estimate from one or more Low Resolution images. Currently, SR is an active research field; a review of SR algorithms is available in while performance comparisons can be found in. In general, SR algorithms can be divided in single-frame (SF-SR) or multi-frame (MF-SR) approaches. The later exploits the motion between successive LR frames in order to extract unique information from each representation.

A sequential approach is used for motion estimation using optical flow between current and past frames while the frame deblurring and blur estimation is performed using an iterative multi-scale approach with a Huber-norm based cost function minimized using the conjugate gradient method. We propose to strengthen the modeling of the SR problem using all available information in the compressed bitstream. In particular, we know the reconstruction levels and the associated quantization intervals for each quantized coefficient in the bitstream.

2.2 PROPOSED SYSTEM:

Using a variational formulation based on a modern proximal dual splitting algorithm, we efficiently combine multiple descriptions of the same video. Two applications are proposed: combining two compressed Low Resolution descriptions of a video sequence into a High Resolution description and enhancing a compressed HR video using a LR compressed description.

Tests are performed over multiple video sequences encoded with High Efficiency Video Coding, at different compression levels and resolutions obtained through multiple down-sampling methods.

2.3 ADVANTAGES:

The constraint will help in balancing the solution. However, when a more complex method is used to provide a better initialization, this constraint might not take advantage of the additional information. For instance, example based super resolution methods introduce new information not contained in the observation due to their learning process.

In this case, the above constraint will not take advantage of this, as it assumes that H is the "proper" way to reverse L and the new information is regarded as a distortion and corrected. Therefore, when using an initialization based on a complex super resolution algorithm rather than a filter based up-sampling this constraint should be disabled.

However, using such a formulation will introduce nonlinear operators which complicate the optimization problem. Furthermore, the coefficients of the residual are computed with respect to a certain prediction at the decoder side. Using a different prediction, albeit a better one might lead to overall worse results when summing with the residual.

2.3 DISADVANTAGES:

In the following section, we will further detail some properties of the proximity operators which are used in this work, followed by a description and discussion of the proposed algorithm.

The learning based methods have no prior knowledge of the compression and degradation models. As such, for fairness of comparison we perform an additional experiment.

The best performing reference method is Vnet. We consider a best case scenario in which we have access to the degraded and original video sequences and the network can be fine-tuned on this particular dataset.

It should be noted, that for a general-purpose training, large video databases should be used without overlapping between training and test data. However, in this experiment we aim to maximize the performance of the network on our test sequences.

CHAPTER 3

INTRODUCTION TO DIGITAL VIDEO PROCESSING

3.1 DIGITAL VIDEO PROCESSING

A digital video is a moving picture, or movie. Digital video processing is the study of algorithms for processing moving videos that are represented in digital format. Here, we distinguish digital video from digital still videos, which are videos that do not change with time basically, digital photographs.

Digital videos are multidimensional signals, meaning that they are functions of more than a single variable. Indeed, digital video is ordinarily a function of three dimensions – two in spaces and one in time, as depicted in Fig. Because of this, digital video processing is data intensive: significant bandwidth, computational, and storage resources are required to handle video streams in digital format.

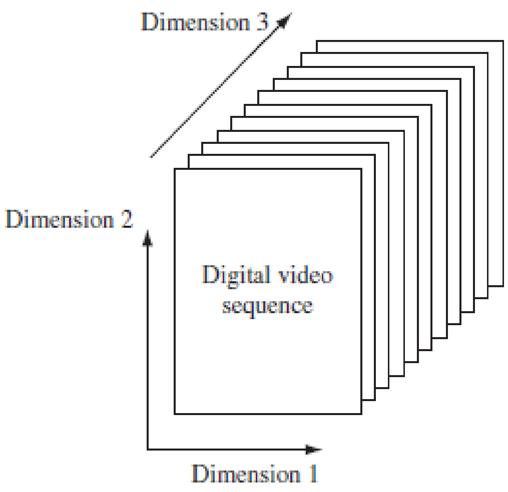

FIGURE 3.1

Digital processing of video requires that the video stream be in a digital format, meaning that it must be sampled and quantized.

HISTORY

The history of video processing starts when image scanning, the fundamental principle enabling television, was first described in 1840's by Bakewell. It lasted almost till the end of the 19th century before Nipkov, in 1883, converted the principle into the first practical system using mechanical scanning. Around 1926, Baird in London and Jenkins in Washington independently gave the first demonstrations of actual television, using Nipkov's invention. By 1935, the electronic scanning system, invented by Braun in 1897, has become mature for the receiving end of the television system and devices implementing transmission standards have been developed. The British Broadcasting Company (BBC) started the first regular black and white TV broadcast in 1936, but it was only after world War II that television broadcasting became popular.

In 1950, a colour television transmission system was developed in the United States, which leads to the NTSC-standard (National Television Sub-Committee) and US color TV broadcasting in 1953. The European alternatives, SECAM (Sequential Color and Memory) developed in France and PAL (Phase Alternation Line) developed in Germany, overcome the sensitivity for phase errors in the transmission path of the NTSC-system. Broadcast based on SECAM and PAL has been in use since 1967. The early black and white TV channels occupied approximately 6 to 8 MHz transmission bandwidth. To introduce the color signal in a compatible fashion, a sub-carrier was added within the least visible part of the video spectrum, modulated by the bandwidth-reduced chrominance signal. By 1970 regular color TV broadcasts had started in most European countries. After over half a century of using analog television systems, a revolution triggered by digital technology is changing the area of video processing profoundly.

The digital technology generates many new applications, particularly because it significantly simplifies storage and delay of signals, but also creates new problems. Also the introduction of personal computers, game consoles and digital recording provide new platforms for digital video and generate new challenges for video processing research.

More recently, revolutionary developments in display technology, where the traditional scanning electron beam hitting an electro-luminant screen, is replaced by a flat array (matrix) of cells that individually generate or modulate light, provide large area, high resolution, bright displays with a perfect geometry unequalled by the earlier display technology. The performance of these displays sets new goals for video processing. In the following, we shall first review some technological trends and application domains and reveal the most interesting challenges in modern video systems. Given the scope of the thesis, we shall focus on challenges in the area of image enhancement.

Trends in video

Digital signal processing did not instantaneously change the video chain, but rather faded into the television receiver with islands of digital signal processing for separate functions to improve the picture quality, such as noise reduction, scan rate conversion, etc. or extending the functionality such as Picture-in- Picture (PIP) and Internet browsing. To date, the process has not ended yet, as this would imply that digital processing should cover every aspect including capture, transmission, storage, etc. Clearly, the impact of digital television is more significant than simply moving from an analog to a digital transmission system. It permits a number of program elements, including video, audio and data that match the services selected by the consumer. The developments in modern display technology, semiconductor manufacturing and digital communication networks, stimulate the evolution of the digital television system and make it feasible.

The evolution from analog to digital

With the existing analog television broadcasting system, including NTSC, PAL and SECAM, the video audio and some limited data information are conveyed by modulating a Radio Frequency (RF) carrier. Digital transmission has advantages over the analog system in many aspects, like stability, ease of integration, flexibility and high quality, while digital video signals can be relayed over a long distance at extremely low bit error rate. However, there are drawbacks as well.

Challenges in digital video processing

Considering the trends discussed in the previous section, various challenges in the area of digital video processing are apparent.

3.2 DIGITAL VIDEO TECHNOLOGY

Video is the technology of electronically capturing, recording, processing, storing, transmitting, and reconstructing a sequence of still videos representing scenes in motion. Digital video, and its associated technology, presents some significant advantages over analogical solutions mainly due to its potential and capability to become just another type of data that can be manipulated, stored and transmitted along with other types of digital data. Consequently, digital video is being adopted in a proliferating array of new applications, like mobile phones, DVD and digital TV, aimed to full the new and more demanding consumer needs. However, despite all the advantages that digital video technology offers, it has also some drawbacks, such as the amount of information that must be transmitted for good quality video communications.

From the most simplistic point of view, a digital video signal can be seen as a sequence of bi-dimensional still videos which are taken at fixed time intervals. Each of these videos is composed by a set of points (or pixels) which represent the visual luminescence of the captured scene at a specific image point. Therefore, a monochromatic digital image can be seen as a 2-D matrix filled with pixel values, which are usually, quantized using an 8-bit representation. On the other hand, since polychromatic videos are made of three different videos, each of them regarding to one of the three primary colors, red (R), green (G) and blue (B), such videos require 24-bit per pixel to represent the scene luminescence. Hence, considering a video sequence at 30 fps with a spatial resolution of 720 X 480 pixels, which are common values for digital television, this implies a signal bit rate of 248:83 Mbits/s, which is too high for typical communication channels. Consequently, digital video signals need to be compressed so that they can be used in practical applications.

Digital Video Processing:

- Digital: The data in desecrate format.
- Video: The sequence of still videos

- Processing: Computing operations(e.g., compression, filtering, retrieval)

Digital VIDEO processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of VIDEO processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation. There are no clear-cut boundaries in the continuum from VIDEO processing at one end to complete vision at the other. However, consider three types of computerized processes low-, mid-, & high-level processes. Low-level process involves primitive operations such as VIDEO processing to reduce noise, contrast enhancement & VIDEO sharpening. A low- level process is characterized by the fact that both its inputs & outputs are VIDEOs.

Mid-level process on VIDEOs involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are VIDEOs but its outputs are attributes extracted from those VIDEOs.

Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in VIDEO analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Video Basics: A video (file types like AVI and MPEG) is a collection of several videos. Each image is called frames and the amount of videos shown per second is called frames per second or simply FPS. The more frames per second your video has, the better, since more realistic the image will be. Videos are usually saved using at least TV quality settings, i.e. 30 frames per second.

3.3 VIDEO AND IMAGE FORMAT:

The format is used to specifies how the image /movie stored in a memory

File format:

The way a computer or digital A/V device stores information any kind of information is through a series of 0's and 1's. Each 0 or 1 is known as a bit; a string of 8 bits makes up a unit of memory known as a byte. A file format is simply a specified way of arranging a given type of information into bytes.

What's a Video Format?

Video formats are confusing because most video files have at least two different types: the container and the codec used inside that container. The container describes the structure of the file: where the various pieces are stored, how they are interleaved, and which codec's are used by which pieces. It may specify an audio codec as well as video. A codec ("coder/decoder") is a way of encoding audio or video into a stream of bytes.

Video Formats:

AVI: audio video interleave, this is basic and system supported format, it has maximum information to form a video sequence.

Video on PC's is cool - you can download video files off the Web, and you capture clips on your home PC to share with friends. There's a standard audio / video file format under Windows called AVI (Audio Video Interleave).

Windows comes with some basic codecs built-in (and with additional ones in more recent versions). If you buy video capture hardware like a USB camera or a PCI board, it will include the codec's needed to understand the formats produced by the hardware. If you buy a video editing program, it will often include additional codec's to support a wider variety of video formats. However, this means you now have a license to create files that other people can't play. Unless they have the same codec, the file is useless to them.

The result is that sharing video files on PC's can be quite messy. Sometimes the hardware or software product provides a way to share codec's with others, or has posted them on a Web site. Sometimes you can find a third party that provides a compatible codec. And sometimes you can't, and it seems like nobody has considered the possibility that you might want to share your video with others. There is no central clearinghouse for video formats, no standard way to figure out how to get an AVI file to play on your system.

The same basic idea of a general file format with add-in codec’s for different video formats is used for the Apple QuickTime video format.

MPEG the Moving Picture Experts Group develops standards for digital video and digital audio compression. It operates under the auspices of the International Organization for Standardization (ISO). The MPEG standards are an evolving series, each designed for a different purpose.

To use MPEG video files, you need a personal computer with sufficient processor speed, internal memory, and hard disk space to handle and play the typically large MPEG file (which has a file name suffix of .mpg). You also need an MPEG viewer or client software that plays MPEG files. (Note that .mp3 file suffixes indicate MP3 (MPEG-1 audio layer-3) files, not MPEG-3 standard files.) You can download shareware or commercial MPEG players from a number of sites on the Web.

Digital video is part of digital versatile disc (DVD), a new optical disc technology that is expected to rapidly replace the CD-ROM over the next few years. The DVD holds 4.7 gigabytes of information on one of its two sides, or enough for a 133-minute movie. With two layers on each of its two sides, it will hold up to 17 gigabytes of video, audio, or other information. (Compare this to the current CD-ROM disc of the same physical size, holding 600 megabytes. The DVD can hold more than 28 times as much information.)

The DVD player will also play regular CD-ROM discs. DVDs can be recorded in any of three formats, variously optimized for: video (for example, continuous movies) audio (for example, long-playing music), or a mixture (for example, interactive multimedia presentations). The DVD drive has a transfer rate somewhat faster than an eight-speed CD-ROM player.DVD uses the MPEG-2 file and compression standard. MPEG-2 videos have four times the resolution of MPEG-1 videos and can be delivered at 60 interlaced fields per second where two fields constitute one image frame. (MPEG-1 can deliver 30 non interlaced frames per second.) Audio quality on DVD is comparable to that of current audio compact discs.

The MPEG standards are an evolving set of standards for video and audio compression and for multimedia delivery developed by the Moving Picture Experts Group (MPEG). MPEG-1 was designed for coding progressive video at a transmission rate of about 1.5 million bits per second. It was designed specifically for Video-CD and CD-i media. MPEG-1 audio layer-3 (MP3) has also evolved from early MPEG work.MPEG-2 was designed for coding interlaced videos at transmission rates above 4 million bits per second. MPEG-2 is used for digital TV broadcast and DVD. An MPEG-2 player can handle MPEG-1 data as well.

MPEG-1 and -2 define techniques for compressing digital video by factors varying from 25:1 to 50:1. The compression is achieved using five different compression techniques:

1. The use of a frequency-based transform called Discrete Cosine Transform (DCT).
2. Quantization, a technique for losing selective information (sometimes known as lossy compression) that can be acceptably lost from visual information.
3. Huffman coding, a technique of lossless compression that uses code tables based on statistics about the encoded data.
4. Motion compensated predictive coding, in which the differences in what has changed between an image and its preceding image are calculated and only the differences are encoded.
5. Bi-directional prediction, in which some videos are predicted from the pictures immediately preceding and following the image.

The first three techniques are also used in JPEG file compression.

A proposed MPEG-3 standard, intended for High Definition TV (HDTV), was merged with the MPEG-2 standard when it became apparent that the MPEG-2 standard met the HDTV requirements.MPEG-4 is a much more ambitious standard and addresses speech and video synthesis, fractal geometry, computer visualization, and an artificial intelligence (AI) approach to reconstructing videos. MPEG-4 addresses a standard way for authors to create and define the media objects in a multimedia presentation, how these can be synchronized and related to each other in transmission, and how users are to be able to interact with the media objects. MPEG-21 provides a larger, architectural framework for the creation and delivery of multimedia. It defines seven key elements:

- Digital item declaration
- Digital item identification and declaration
- Content handling and usage
- Intellectual property management and protection
- Terminals and networks
- Content representation
- Event reporting

The details of various parts of the MPEG-21 framework are in various draft stages

CHAPTER 4
MODULE DESCRIPTIONS

4.1 VIDEO CONSTRUCTION-

Current solutions do not take into account the particularities of different video encoders, while video reconstruction methods from compressed sources do not provide resolution enhancement.

In this paper, we propose a multi-source compressed video enhancement framework where each description can have a different compression level and resolution. Using a variational formulation based on a modern proximal dual splitting algorithm, we efficiently combine multiple descriptions of the same video.

4.2 SUPER RESOLUTION-

Super-resolution (SR) algorithms are post-processing techniques that infer a spatially High Resolution (HR) estimate from one or more Low Resolution (LR) images. Currently, SR is an active research field; a review of SR algorithms is available in while performance comparisons can be found.

In general, SR algorithms can be divided in single-frame (SF-SR) or multi-frame (MF-SR) approaches. The later exploits the motion between successive LR frames in order to extract unique information from each representation.

For the super-resolution case, the sole TV constraint may be insufficient. In particular, we compute data fidelity w.r.t. the reconstruction levels only using the LR (and transformed) versions of the solution. As a matter of fact, among all the possible solution providing the desired minimum distance, there is still no guaranteeing that unlikely ones will not be picked.

Super-resolution imaging is a class of techniques that enhance (increase) the resolution of an imaging system. In optical SR the diffraction limit of systems is transcended, while in geometrical SR the resolution of digital imaging sensors is enhanced.

Super-resolution imaging techniques are used in general image processing and in super-resolution microscopy.

Because some of the ideas surrounding super-resolution raise fundamental issues, there is need at the outset to examine the relevant physical and information-theoretical principles.

4.3 HEVC

We integrate HEVC compression model and finally we implement the framework using a modern and efficient proximal dual-splitting algorithm such that we can combine observations with different compression levels, down-sampling methods or resolutions. Furthermore, this approach provides a great degree of flexibility. The framework can be applied to various applications ranging from image SR to enhancing the quality and suppressing compression artifacts of a video, and even combining multiple video streams, as shown in our proposed applications.

High Efficiency Video Coding (HEVC), also known as H.265 and MPEG-H Part 2, is a video compression standard designed as part of the MPEG-H project as a successor to the widely used Advanced Video Coding (AVC, H.264, or MPEG-4 Part 10). In comparison to AVC, HEVC offers from 25% to 50% better data compression at the same level of video quality, or substantially improved video quality at the same bit rate. It supports resolutions up to 8192×4320, including 8K UHD, and unlike the primarily 8-bit AVC, HEVC's higher fidelity Main 10 profile has been incorporated into nearly all supporting hardware.

4.4 INFORMATION

When the term super-resolution is used in techniques of inferring object details from statistical treatment of the image within standard resolution limits, for example, averaging multiple exposures, it involves an exchange of one kind of information (extracting signal from noise) for another (the assumption that the target has remained invariant).

4.5 RESOLUTION:

True resolution involves the distinction of whether a target, e.g. a star or a spectral line, is single or double, ordinarily requiring separable peaks in the image. When a target is known to be single, its location can be determined with higher precision than the image width by finding the centroid (center of gravity) of its image light distribution. The word ultra-resolution had been proposed for this process but it did not catch on, and the high-precision localization procedure is typically referred to as super-resolution.

The technical achievements of enhancing the performance of imaging-forming and –sensing devices now classified as super-resolution utilize to the fullest but always stay within the bounds imposed by the laws of physics and information theory.

CHAPTER 5

SOFTWARE INTRODUCTION

5.1 INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping1
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems.

5.2 THE MATLAB SYSTEM:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

Handle Graphics. This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API). This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

5.3 DEVELOPMENT ENVIRONMENT

Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.

On a UNIX platform, to start MATLAB, type matlab at the operating system prompt.

After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.

The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for

Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who's.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Non-functional requirement

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. Non-functional requirements add tremendous value to business analysis. It is commonly misunderstood by a lot of people. It is important for business stakeholders, and Clients to clearly explain the requirements and their expectations in measurable terms. If the non-functional requirements are not measurable then they should be revised or rewritten to gain better clarity. For example, User stories help in mitigating the gap between developers and the user community in Agile Methodology.

Usability:

Prioritize the important functions of the system based on usage patterns. Frequently used functions should be tested for usability, as should complex and critical functions. Be sure to create a requirement for this.

Reliability:

Reliability defines the trust in the system that is developed after using it for a period of time. It defines the likeability of the software to work without failure for a given time period.

Create a requirement that data created in the system will be retained for a number of years without the data being changed by the system.

It's a good idea to also include requirements that make it easier to monitor system performance.

Performance:

Think of stress periods, for example, at the end of the month or in conjunction with payroll disbursement.

Supportability:

The system needs to be cost-effective to maintain.

Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documentation, e.g. which test cases and test plans will accompany the system.

CHAPTER 6

SOURCE CODE

```
clc
clear all
close all

[filename, pathname]=uigetfile ('*.avi');

str2='.jpg';
file=aviinfo (filename); % to get information abt video file
frm_cnt=file.NumFrames ;          % No.of frames in the video file

for i=1:frm_cnt
    frm (i) =aviread (filename, i);          % read the Video file
    frm_name=frame2im (frm (i));
%   frm_name=rgb2gray (frm_name);       % Convert Frame to image file
   filename1=strcat (strcat (num2str (i)), str2);
    imwrite (frm_name, filename1);
end

%fprintf ('please wait....');
    filebase = dir ('*.jpg');                    % If you are using a
continuous frame of images, start from here.

    num_files = numel (filebase);
    images = cell (1, num_files);
     AS=cell (1, num_files);

     for k = 1:num_files
         images {k} = imread (filebase (k).name);
         [rows columns color]=size (images {1});
         if (color==3)
             AS {k} =rgb2hsv (images {k});
             A (k) =im2frame (AS {k});
         else
             AS {k} = gray_level_images (images{k});

         end
     end

     MS=cell (1, num_files);

     for j=1:frm_cnt
 filename_1=strcat (strcat (num2str (j)), str2);
 X=imread (filename_1);
S=size(X);
k=1;
l=1;
for m=1: S (1)
     for n=1: S (2)
         if X (m, n) <=60
%                X (m, n) =0;
               Y (k, l) =X (m, n);
         end
```

```
    end
 end

for k=1:frm_cnt
    cimg=strcat (num2str (k), str2);
    v=imread (cimg);
    [Y, map] = rgb2ind (v, 255);
     F (:,:, k) =im2frame (flipud(Y), map);

end
end

     for k = 1:num_files
         images {k} = imread (filebase (k).name);
         [rows columns color]=size (images {1});
         if (color==3)
             MS {k} =image_enhancement_sw (images {k});
             M (k) =im2frame (MS {k});
         else
             MS {k} =gray_level_images (images {k});

         end
     end

%movie (M)

mov = aviread (filename);
%movie (mov)
%figure;
%movie (F)

[h, w, p] = size (mov (1).cdata);
hf = figure;
set (hf, 'position', [150 150 w h]);
movie (gcf, mov);

[h, w, p] = size (A (1).cdata);
hf = figure;
set (hf, 'position', [150 150 w h]);
movie (gcf, A);

[h, w, p] = size (M (1).cdata);
hf = figure;
set (hf, 'position', [150 150 w h]);
movie (gcf, M);

input_file_size = frm_cnt * size (frm (1).cdata, 1)* size (frm (1).cdata,
2) * size (frm (1).cdata, 3)

%output_file_size=frm_cnt * size (F (1).cdata, 1)* size (F (1).cdata,2) *
size(F(1).cdata,3)

mse=(sum(mov(1).cdata(:,:,1)-F(1).cdata).*sum(mov(1).cdata(:,:,1)-
F(1).cdata))/input_file_size
 psnr=20*log10 (255/sqrt (min (mse))
```

CHAPTER 7

SIMULATION RESULTS

Figure 7.1 Input video

Description: Input video with blur.

From the figure 7.1, We begin by discussing the experimental setup and defining the test bench architecture and algorithm parameters which are used throughout main experiments. A series of small tests are performed in order to show proposed method's behaviour and motivate the framework set-up for the two proposed applications.

Two observations are generated by applying two different degradation operators denoted by $L1$ and $L2$ followed by a compression step with a video coder. The parameter accounts for the unequal quality of the observations. This parameter may not be easily estimated, since the quality of the observations is not measurable with respect to the unavailable original sequence at the decoder side

Figure 7.2 Noised Video

Description: - From the figure 7.2, The Soft noise is added to the input video. Video is not clear and quality of the video is not good, which leads to loss of information.

Figure 7.3 Reconstructed Video

Description: From the figure 7.3, The Noise is removed from the input video, which leads to get clear information from the video.

CHAPTER 8

CONCLUSION

This work presents a model-based SR approach specifically designed for compressed video streams, and focuses on scenarios where multiple observations are available. The proposed model makes an explicit use of the available compressed syntax (encoded coefficients, unit sizes, etc.) and builds a heterogeneous cost function combining data-fidelity objectives and a priori constraints. The resulting minimization problem, efficiently solved via convex optimization, embeds the SR result into a domain that closely fits the given compressed observations. Experimental results demonstrate that in most cases combining the complementary information available in the different observations allows very efficient SR, significantly outperforming the capabilities of image or single video stream. Indeed, quality improvements superior to 2dB w.r.t. one of the best performing learning-based single image SR method can be observed for high-quality encodings, which has a noticeable impact on the visual quality of the reconstructed video sequence. Extending the framework application to other compression schemes (AVC, JPEG, JPEG2000, VC9, etc.) is straightforward. Another interesting future research direction is to combine AVC and HEVC video streams. Yet, short-term research focuses on investigating more thoroughly the complexity and real-time capabilities of the proposed framework, requiring the implementation and optimization of the convex solver on parallel processing platforms.

REFERENCES

[1] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, “Overview of shvc: Scalable extensions of the high efficiency video coding standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, pp. 20–34, 2015.

[2] A. Purica, E. G. Mora, M. Cagnazzo, B. Pesquet-Popescu, and B. Ionescu, “Multiview plus depth video coding with temporal prediction view synthesis,” IEEE Transactions on circuits and systems for video technology, vol. 26, no. 2, pp. 360–374, 2016.

[3] P. Milanfar, Super-Resolution Imaging, Digital Imaging and Computer Vision. Taylor & Francis/CRC Press, 2010.

[4] S. K. Nelson and A. Bhatti, “Performance evaluation of multi-frame super-resolution algorithms,” in International Conference on Digital Image Computing: Techniques and Applications (DICTA), Centre for Intelligent Systems Research, Deakin University, Geelong , Victoria ,Australia, 2012.

[5] L. Tian, A. Suzuki, and H. Koike, “Task-oriented evaluation of superresolution techniques,” in nt. Conference on Pattern Recognition (ICPR), 2010, pp. 493–498.

[6] Y. J. Kim, J. H. Park, G. S. Shin, H.-S. Lee, D.-H. Kim, S. H. Park, and J. Kim, “Evaluating super resolution algorithms,” in SPIE-IS&T Electronic Imaging, Image Quality and System Performance VIII, vol. 7867, 2011.

[7] Z. Jin, T. Tillo, C. Yao, J. Xiao, and Y. Zhao, “Virtual-view-assisted video super-resolution and enhancement,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 3, pp. 467–478, March 2016.

[8] S. Babacan, R. Molina, and A. Katsaggelos, “Variational bayesian super resolution,” IEEE Transactions in Image Process, pp. 984–999, 2019..

[9] R. Molina, J. Nez, F. Cortijo, and J. Mateos, “Image restoration in astronomy bayesian perspective,” IEEE Signal Process. Mag., vol. 18, pp. 11–29, 2001. S. Villena, M. Vega, D. Babacan, R. Molina, and A. Katsaggelos, “Bayesian combination of sparse and non sparse priors in image super resolution,” Digital Signal Processing, vol. 23 (2), pp. 530–541, 2014.