Performance Improvement and Energy Saving Solutions In Phase Unwrapping and Video Communication Applications

by

Bardia Barabadi
B.Sc. (Hons), Sharif University of Technology, Iran, 2018

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Performance Improvement and Energy Saving Solutions In Phase Unwrapping and Video Communication Applications

by

Bardia Barabadi

B.Sc. (Hons), Sharif University of Technology, Iran, 2018

Supervisory Committee

Dr. Amirali Baniasadi, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Nikitas J. Dimopoulos, Departmental Member

(Department of Electrical and Computer Engineering)

ABSTRACT

In the form of images and videos, visual content has always attracted considerable interest and attention to itself since the early days of the computer era. Although, due to the high density of information in such contents, it has always been challenging to generate, process and broadcast videos and images. These challenges grew along with the demand for higher quality content and attained the research community's attention to themselves. Even though many works have been done by researchers and engineers in academic and industrial environments, the demand for high-quality content introduces new constraints on the quality, performance (speed) and energy consumption. This thesis focuses on a couple of image and video processing applications and introduces new approaches and tweaks to improve the performance and save resources while keeping the functionality intact.

In the first part, we target Interferometric Synthetic Aperture Radar (InSAR), an imaging technique used by satellites to capture the earth's surface. Many algorithms have been developed to extract useful information, such as height and displacement, from such images. However, the sheer size of these images, along with the complexity of most of these algorithms, lead to very long processing time and resource utilization. In this work, we take one of the dominant algorithms used for almost every In-SAR application, Phase Unwrapping, and introduce an approach to gain up to 6.5 times speedups. We evaluated our method on InSAR images taken by the Radarsat-2 sensor and showed its impact on a real-world application.

In the second part of this thesis, we look at a prevalent application, video streaming. These days video streaming dominates the internet traffic, so any slight improvement in terms of energy consumption or resource utilization will make a sizable difference. Although the streamers use various encoding techniques, the quality of experience

of the clients prevents them from overplaying these techniques. On the other hand, there has been a growing interest in another venture of research which focuses on developing techniques that aim to restore the quality of the videos that have been subjected to compression. Although these techniques are used by many users on the receiver side, the streamers often ignore their capabilities. In our work, we introduce an approach that makes the streamer aware of the capabilities of the receiver and utilizes that awareness to reduce the cost of transmission without compromising the end user's quality of experience. We demonstrated the technique and proved our concept by applying it to the HEVC encoding standard and JCT-VC dataset.

Contents

List of Figures

ACKNOWLEDGEMENTS

I would like to thank my supervisor **Prof. Baniasadi** for mentoring, encouragement, and patience; **Prof. Dimopoulos** for his insights and guidance throughout my MASc period. Furthermore I would like to thank **my parents** who supported me from far far away. Also, many thanks to my significant other, my bike and beautiful British Columbia for supporting me in low moments.

We become what we think about all day long

R. Waldo Emerson

To my parents.

Chapter 1

Introduction

From the early days of the computer era, humans were thirsty for visual content in the form of images and video. However, there always were multiple barriers and difficulties in producing, broadcasting and processing such content. During the last years of the 20^{th} century, the internet was introduced to the general public. It did not take long for the curious users to explore and scrutinize this world-changing innovation. In just a few years, the internet became full of images, videos travelling through the world wide web. Of course, most of these visual contents were as big as a postage stamp, and it would take minutes to download. The need for a great change was evident. It was up to researchers in academia and the industry to come up with ideas to compress, transfer and process the images and videos so that we can enjoy them on our huge screens in a blink of an eye, as we do today. Of course, the quest for speed, quality and efficiency is not over. With growing technology came a growing demand for faster speeds and higher resolutions from even a more significant number of consumers. This journey brought us here, to an era where many lives depend on communications. Many decisions are based on information extracted from enormous

data collected by satellites orbiting the earth. This work is a small brick added to the skyscraper of technology that I hope someday takes humanity to prosperity and beyond.

In this thesis, we take a couple of applications from two different worlds and redesign them to get the same functionality while saving energy and resources. In both works, an out-of-the-box approach is introduced to provide considerable improvements. All claims are then supported by qualifying experiments discussed at the end of each chapter. Below is an introduction to each of these applications and ideas.

The first application we focus on is an imaging technique known as Interferometric Synthetic Aperture Radar (In-SAR). This technique is a variant of remote sensing that is used by satellites to map the elevation of the planet. The idea is to send a fixed wavelength signal to the earth and capture the reflection of it from the surface of the earth. Comparing the two waves provides information on the distance that was traversed by the signal. Modern satellites capture high-resolution images by choosing the correct wavelength and high-tech instruments. However, the raw captured images are not usable as-is. They need to go through a complex pipeline of different data processing algorithms to become useful. Although, since the data size in this type of images is extraordinarily huge, performing these computations consumes significant time and energy. In the first section of this work, we aim to optimize the phase unwrapping algorithm, which dominates the processing power of the InSAR analyzing pipeline.

We start with defining the problem of phase unwrapping and explain how it is related to the physical phenomenon process of capturing InSAR images. We show that phase unwrapping (PU) can be characterized as an optimization problem. In section 2.3 we go over the previous research that has been done on solving these optimization

problems efficiently. Section 2.4 is where we take one of these algorithms, Primal-Dual, examine its scalability and sensitivity to problem size, and explain our approach for improving its performance. Finally, in section 2.4.5, we introduce the dual-stage method, a novel approach that breaks up the initial problem into sub-problems that can be solved individually. The results of these sub-problems are then combined to provide an initial state for the parent problem. In section 2.5 we test our proposal by profiling the performance gain over a set of real-world (actual InSAR images) datasets. The outcomes of this chapter are published in IEEE-NORCAS2019.

In chapter 3 of this thesis, we look at a completely different application, that is, video streaming. Every day millions of people use streaming services such as Netflix to watch videos around the world. This usage increased significantly due to the ongoing pandemic [14]. Because of the sheer size of the videos being transmitted on the internet, even the smallest improvement in transmission efficiency would have a considerable impact on the total power and resource consumption. Although many works have been done on reducing the utilized bandwidth and improving the collective transmission speed, few consider thinking outside the box and looking for a missing piece in the puzzle. We noticed that, in a separate venture of research, many papers have been published on the subject of quality enhancement. In the context of this thesis, quality enhancement (QE) refers to the process of restoring the details in an image (or a video frame) after an event of quality loss. This quality loss may be due to compression, downsampling, etc. We noticed that the performance of such QE algorithms are growing rapidly, especially after Machine Learning was added to the researchers' tools. This knowledge sparked an idea: Why is no one using the QEs to save resources and bandwidth, rather than just refining the viewer's quality of experience? To answer this question, we introduce RAB: A Receiver-Aware Bandwidth Saving Solution. We propose that the transmitter, i.e. the streamer, can

compress and encode the videos more efficiently if it is aware of the existence of a QE on the receiver side. We expanded our proposal in section 3.4 by introducing a framework to quantify the abilities of any QE, and a procedure to act upon when a QE is detected. We quantified our assumptions and results by introducing two assessment measures in section 3.3.2. Then we set up a set of experiments to prove our concept using a certain QE, explained in section 3.6.

Chapter 2

Dual Stage Phase Unwrapping

2.1 Abstract

Energy minimization on large scale problems can be expensive to compute. This chapter presents an extension of a primal-dual algorithm to scale to large scale convex energy minimization problems, while guaranteeing the global optimum on convergence. This is done through the use of a so-called "Dual-Stage" approach that exposes a natural level of parallelism. Benchmarks on the Phase Unwrapping problem in InSAR show that using this approach can give significant improvements to computation-time and memory-time used.

2.2 Introduction

Energy minimization is an established methodology to compute the global minimum of an appropriate optimization problem, especially in computer vision tasks [6]. Typical examples of such tasks may include dense stereo restoration [33], image segmen-

tation [5], multi-view image reconstruction [32], and panoramic stitching [10]. In this work, we are especially interested in those problems that use the minimization of convex energy functions. In this particular case, specialized and more efficient algorithms have been developed [10]. One such problem of particular interest is the task of Phase Unwrapping (PU), which is closely related to that of panoramic image stitching.

While most fields in Computer Vision have experienced an explosive growth in the availability of high-resolution data, in the field of remote sensing, Synthetic Aperture Radar (SAR) data providers have recently launched several missions that resulted in a deluge of available data from all over the world [20]. This has placed a great emphasis on accelerating resource hungry tasks in the field of Interferometric SAR (InSAR) processing, and in particular, on the task of PU.

In this chapter, we focus on a particular Primal-Dual (PrDu) algorithm for solving the convex pairwise Markov Random Field energy minimization problem, and present a simple approach to scale this algorithm to large datasets with a significant improvement in run time and memory-time used. We present a hypothesis to explain the speedup observed and empirical evidence on real benchmarks.

2.2.1 Problem Statement

Certain vision tasks can be stated in terms of a pairwise energy minimization problem, where the energy has the following form,

$$E(x) = \Sigma_{u \in V} D_u(x_u) + \Sigma_{(u,v) \in G} P_{uv}(x_v, x_u). \tag{2.1}$$

Here G is the set of edges and V is the set of nodes. In the context of an image, nodes correspond to the pixels and edges are the relationship between neighboring

pixels as defined by the problem in hand. P are the pairwise terms and D are the unary terms, with x assumed to be integer values. For most vision tasks, the value of x maps to a quantity that it attempts to estimate.

In this work, we are particularly interested in the case where the functions D and P are convex. For certain convex function forms of D and P, there are specialized mappings to an equivalent max-flow problem. Here, we are interested in the $L1$ minimization form where D and P have the form:

$$D_u(x_u) = w_u|x_u - a_u| \text{ and } P(x_v, x_u) = w_{uv}|x_v - x_u - a_{uv}| \tag{2.2}$$

where w, a are integral. In section 2.4.2 we explain how these two terms are defined. For this special form, we focus on applying the PrDu algorithm of [10], and scaling it to large datasets for the task of Phase Unwrapping.

2.2.2 Contribution

To scale the PrDu algorithm to large scale data we propose to find a so-called *estimation* for each variable via simplifying the original problem. We do this simplification by dividing the problem variable space into smaller chunks using a graph partitioning algorithm implemented in the METIS [30] library. Doing this, we find the optimum solution for each segment much faster as the relationship between performance and problem size scales poorly. We show that the run time of the algorithm is highly affected by the initial state that it starts from. By combining these two characteristics the Dual-Stage method gains up to 6.5x speedup while having a 21.4 percent memory overhead in experiments later presented. To summarize, our contributions are:

- Partitioning the Problem: the METIS algorithm is used to find the best partitioning of the problem into smaller sub-problems.

- Dual-Stage: PrDu is used to solve the sub-problems. The results build an estimation for all the variables to be used by the PrDu algorithm, providing a better initialization for the solver.

- InSAR Experiments: Several experiments show the effectiveness of the method for the benchmark of Phase Unwrapping InSAR interferograms.

- Speedup Analysis: Investigating the obtained speedups and presenting a working hypothesis to explain the observed speedup.

2.2.3 Chapter Organization

This chapter proceeds as follows. In the next section, we review the Phase Unwrapping problem and existing solvers. In section 2.4 we explain our motivation and methodology and in section 2.6 we discuss and justify our results.

2.3 Background

2.3.1 InSAR

Synthetic Aperture Radar (SAR) is a type of radar which can be used as an active remote sensing system. SAR measures the phase return of the electromagnetic pulse radiated from the antenna and its reflection from targets that are imaged on the ground [9]. Each SAR image is a complex function on a 2-D grid containing the phase information obtained from the electromagnetic signal that is reflected from targets. This image format is called Single Look Complex (SLC) [46]. The interferometric [37]

difference between two independent SLCs, imaging the same targets, incorporates information on terrain elevation and target deformation mixed in a superposition of several other signals, each imprinted on the phase data collected. Typically, to detect surface deformation or land subsidence, the SAR images are taken every few days (usually around 10-20 days between successive acquisitions.) This imagery method is called SAR interferometry, or InSAR [19]. A typical goal of using InSAR is to extract a reliable time series of deformation from the data. The topographic signal in the phase can be used to position the targets in 3-D space so the deformation can be localized accurately. However, there are many contaminants such as atmospheric signal which makes the analysis costly and complicated.

2.3.2 Phase Unwrapping

When working with InSAR data, we are dealing with a phase delay signal, hence the measured value only takes values in a wrapped domain. Although the original phase signal can take on any real value, the measurements can only observe modulo 2π of the original signal. This process of wrapping introduces unwanted artificial jumps in the data, and a loss of information. In equation (2.3) below we define the wrapping operator, $W(.)$ [40].

$$W(x) = arg(e^{ix}) = [(x + \pi) \bmod 2\pi] - \pi \tag{2.3}$$

The problem of phase wrapping is used in many remote sensing and medical imagery applications such as InSAR, MR-Angiography and MR-Venography [38]. To recover the original, smooth (with no 2π jumps) phase, an unwrapping algorithm is required. In the process of PU, the objective is to obtain the unwrapped signal, which carries information about the superposition of all signals. The goal is to find an integer value

(K) for each pixel which corresponds to the (relative) integer number of wavelengths the microwave signal cycles through on its journey from the sensor to the reflective target and back. This quantity is referred to as the phase ambiguity of each pixel or for the sake of brevity, the ambiguity. Finding the absolute ambiguity (exact value of K) is impractical, if not impossible, since there is no information on the absolute number of missing cycles in the phase difference values. The information that we try to find is not the absolute value but the relative value of the measured quantity (distance) between the pixels in the scene. Once complete, phase unwrapping reduces the complexity of any subsequent modelling.

2.3.3 Phase Unwrapping Methods

Phase Unwrapping is a complicated problem to solve, as the information lost in the process of wrapping (number of cycles), cannot be uniquely recovered [21]. As such, our goal is to develop a robust technique to integrate all pertinent prior knowledge and models that explain the observed phase data. However, in the absence of further information, the Phase Unwrapping problem becomes an ill-posed one. Therefore, most Phase Unwrapping formulations fallback on the use of a simple prior on the sampling rate of the sensor. The underlying assumption on the sampling rate is known as the Nyquist Criterion[21].

In the Nyquist Criterion, the sampling rate of the sensor (resolution in case of InSAR) is assumed to be high enough to prevent aliasing from occurring prior to the wrapping process. In other words, before wrapping, every two neighbouring pixels differ by a maximum of a half-cycle, π. Under the Nyquist assumption, if i and j indicate adjacent samples (pixels), we have:

$$\varphi_i - \varphi_j < \pi \tag{2.4}$$

This, also known as *Itoh's condition*, provides us with an estimate of the unwrapped phase gradient. In the case of 1-D signals, we can easily evaluate the derivatives (differences) among contiguous samples and assign integer values such that these discontinuities ($\geqslant \pi$ jumps) do not exist[28].

The problem is more complicated for 2-D signals. The complexity arises when the Nyquist assumption is violated. This can happen in two cases. First, if there is a discontinuity in the absolute phase (ϕ), caused by an underlying process in the scene (e.g., a fault caused by an earthquake). Under such circumstances, the criterion fails due to insufficient sampling[42]. The second scenario occurs if noise corrupts the data [3]. Addressing these challenges is beyond the focus of this work (refer to [36, 8, 17, 2]).

In practice, it is common for some large spatial regions of data to be dominated by noise while others are less so[1]. The noisy regions sometimes prevent the rest of the scene to be unwrapped properly. In the case of noisy patches of the data, neighbouring data points do not obey the Nyquist criterion. To overcome this problem, a common technique is to use only a subset of data which has a high signal to noise ratio (SNR). This selection is usually based on correlation maps [41], which often leave a sparse set of points for phase unwrapping. Instead of matching neighbouring pixels, we use other methods such as Delaunay triangulation to find pixel pairs which have a high SNR and are more likely to obey the criterion [12].

In general, practitioners use two main techniques for Phase Unwrapping i.e., local and global [18]. Local methods integrate estimations over a path which passes through all points. In practice, these algorithms turn out to be sensitive to the choice of the path and therefore their solutions can vary. Global methods, however, build up

[1]Vegetation and water bodies show up as noise in the SAR images.

a model which simultaneously account for all provided estimates (refer to [23, 22, 13]). In this work we use the graph-cut methods for PU. Using the Graph-Cuts approach, the Phase Unwrapping cost function (equation 2.2) is represented as a second-order Markov Random Field. J. Bioucas-Dias introduced this framework in [1]. Bioucas showed that the exact solution to this energy minimization problem could be achieved by finding a finite sequence of binary minimizations, each of which results in an increment to a subset of variables. As shown in [33] and [7], these binary minimizations can be solved by solving max-flow/min-cut on a given graph.

Path-Following Methods

[23] proposed a 2-D Phase Unwrapping model building on top of the Nyquist criterion. They calculate the sum of phase differences over each set of four (squared) pixels in a clockwise manner. If:

$$|\oint \nabla\phi \cdot dr| > 2\pi$$

then it is called a *residue*. In this equation $\nabla\phi$ represents the phase difference between the pixels on the square. The polarity of the residue can be positive or negative. The following step is finding *Branch-Cuts* connecting these residues such that no net residue can be encircled. Branch-Cuts are artificial barriers that introduce $\pm 2\pi$ discontinuities between the samples on opposite sides. The placement of the cuts represents the corresponding Phase Unwrapping solution. However, minimizing the length of the *Branch-Cuts* is an NP-hard problem [23]. The notion of a residue limits this approach to work only in the case of orientable surfaces, graphs or planar graphs.

Minimum Norm Methods

Another class of Phase Unwrapping methods are the the Least-squares methods or more generally minimum L^p-norm methods. Here, $\phi_{i,j}$ and $\varphi_{i,j}$ denote the wrapped and unwrapped phases at (i, j) location respectively. Minimum L^p-norm methods require that the phase differences of the $\phi_{i,j}$ agree with those of the $\varphi_{i,j}$ at the same location in the L^p-norm sense [22]. This means that, $\varphi_{i,j}$ is the minimum L^p-norm solution if it minimizes the following cost function ($W(.)$ is defined as the wrapping function):

$$E = \sum_i \sum_j |\varphi_{i+1,j} - \varphi_{i,j} - W(\phi_{i+1,j} - \phi_{i,j})|^p + \sum_i \sum_j |\varphi_{i,j+1} - \varphi_{i,j} - W(\phi_{i,j+1} - \phi_{i,j})|^p \quad (2.5)$$

Equation 2.5 can be re-arranged to take a form as Equation 2.2 (for $p = 1$). In particular, when all variables in the minimum L1 norm method are integers, the integrality of the solution of the corresponding linear program is guaranteed. This is due to the fact that the linear program can be shown to be totally unimodular [13].

Graph-Cut Methods

The last class of Phase Unwrapping methods involves the use of Graph-Cuts. Using the Graph-Cuts approach, the Phase Unwrapping cost function is represented as a second-order Markov Random Field. J. Bioucas-Dias introduced this framework in [1]. Bioucas showed that the exact solution to this energy minimization problem could be achieved by finding a finite sequence of binary minimizations, each of which

results in an increment to a subset of variables. As shown in [33] and [7], these binary minimizations can be solved by solving max-flow/min-cut on a given graph.

2.4 Dual-Stage Method

2.4.1 Primal-Dual Algorithm

In this work, we use the Primal-Dual algorithm described in [10], which resembles both the minimum norm methods and Graph-Cut methods but has speed advantages over both. Primal-Dual (PrDu), starts from a given initial state, i.e. an estimated value for each variable. In each iteration, PrDu tries to find the best two binary vectors with the same size as the variable vector, tracking both a primal and dual solution in each iteration.

PrDu tracks the *Up* vector, which indicates which variable must increase to take the energy function one step closer to its global minima. The *Down* vector marks the variables which are required to decrement. PrDu finds *Up* and *Down* vectors by constructing a graph by using current configurations and solving a maximum-flow problem. These steps are called *Maxflow-Up* and *Maxflow-Down*. PrDu terminates whenever both *Up* and *Down* contain only zero values indicating that there are no moves that can reduce the energy function.

PrDu builds on the primal algorithm by also tracking a dual solution, which is a valid flow on a somewhat different, but related graph. In the PrDU, the Karush-Kuhn-Tucker conditions are almost all satisfied except for the complementary slackness conditions corresponding to the nodes of the graph. In each iteration, the number of violated conditions is guaranteed not to increase, and convergence is guaranteed after a finite number of iterations.

2.4.2 Cost Function

As mentioned in the previous sections, phase unwrapping is an impossible problem as the unwrapped phase array contains information that is not available in the wrapped array. Therefore, all algorithms rely on some assumptions to perform. The most common of these assumptions is that the Nyquist criterion, i.e. (2.4), is met throughout most (but not necessarily all) of the scene, i.e. the spatial sampling rate is assumed to be high enough that aliasing is avoided. [11]

Here we explain the two terms that we defined in the cost function that we defined for the phase unwrapping problem in section 2.2.1. The final cost function defined in equation (2.1) where D and P are defined in equation (2.2). Below is a more simplified representation of these two equations:

$$E = \sum_{(e,v)\in E} w_{uv}|x_v - x_u - a_{uv}| + \sum_{u\in V} w_u|x_u - a_u|$$

Pairwise Terms The above assumption takes us to the criteria that the true unwrapped phase values of neighboring pixels are assumed to lie within the one-half cycle (π-rad) of each other. The key to phase unwrapping, therefore, lies not on directly estimating the unwrapped phase values themselves, but in estimating the *Unwrapped Phase Differences* between them. This illustrates the fundamental premise of phase unwrapping: Under appropriate conditions, an accurate estimate of the unwrapped solution can be determined from the relationships between neighboring phase samples. Thus, the notion of adequate sampling reduces to the assumption that, locally, the *unwrapped phase differences* are equal to the observed *wrapped phase differences* i.e. the difference in phase wrapped into the interval $[-\pi, \pi)$. This will take us to the following equations. (φ represents the original unwrapped phase, ϕ denotes wrapped

phase array and K represents the label i.e. the ambiguity level per pixel. [11]

We know that:

$$\varphi_u = \phi_u + 2\pi K_u, \varphi_v = \phi_v + 2\pi K_v \tag{2.6}$$

$$\tilde{X} = Wrap(X) = arg(e^{iX}) = X - 2\pi \left\lfloor \frac{X}{2\pi} \right\rfloor \tag{2.7}$$

According to Nyquist Criterion,:

$$\forall (u, v) \in E : \varphi_u - \varphi_v < 2\pi \tag{2.8}$$

By subtracting (2.6) we get:

$$\varphi_u - \varphi_v = \phi_u - \phi_v + 2\pi(K_u - K_v) \tag{2.9}$$

$$\xrightarrow{Wrap} \varphi_u - \varphi_v - 2\pi \left\lfloor \frac{\varphi_u - \varphi_v}{2\pi} \right\rfloor = Wrap(\phi_u - \phi_v) + 2\pi(K_u - K_v) - 2\pi \left\lfloor \frac{2\pi(K_u - K_v)}{2\pi} \right\rfloor$$

From (2.8) we know that $\left\lfloor \frac{\varphi_u - \varphi_v}{2\pi} \right\rfloor = 0$. Also K_u and K_v are integers. So, $\lfloor K_u - K_v \rfloor = K_u - K_v$. That's also the reason why we could break down the wrap function on the right side. By applying the mentioned updated we get:

$$\Rightarrow \varphi_u - \varphi_v = Wrap(\phi_u - \phi_v) \tag{2.10}$$

$$\Rightarrow \phi_u - \phi_v + 2\pi(K_u - K_v) = Wrap(\phi_u - \phi_v)$$

$$\Rightarrow K_u - K_v = \frac{Wrap(\phi_u - \phi_v) - (\phi_u - \phi_v)}{2\pi}$$

We define $\frac{Wrap(\phi_u - \phi_v) - (\phi_u - \phi_v)}{2\pi}$ as a_{uv}

$$\Rightarrow K_u - K_v = \frac{Wrap(\phi_u - \phi_v) - (\phi_u - \phi_v)}{2\pi} \doteq a_{uv}$$

Ideally the equation above should always be correct. Therefore:

$$Objective : \underset{K_u, K_v}{minimize} \mid K_u - K_v - a_{uv} \mid$$

Unary Terms The unary terms is the single summation on the right side of the equation. It helps the minimization algorithm to obtain a unique result by enforcing a constraint. In practical use of phase unwrapping algorithms, this unary terms are being used to provide external information (such as GPS data) to grantee the correctness of the solution.

$$\text{Unary Terms} : \sum_{u \in V} w_u |x_u - a_u|$$

2.4.3 Motivation

As explained earlier, in each step of the PrDu algorithm, each variable is only capable of jumping one step away from its current value. Therefore, the number of steps required to converge to the desired final state can be found using the equation below.

$$MES \equiv max_u\{|K_u^{initial} - K_u^{final}|\} \tag{2.11}$$

For simplicity, we call this quantity "Maximum Error Span" or MES for short. Conventionally, the initial state for the Primal-Dual algorithm is set to be all-zero. The PrDu algorithm iteratively takes *Up* and *Down* steps to take each variable to its final value. We assume that the final values for all of the variables are limited to $[-K_x, K_x]$

range[2] as defined below:

$$\text{if } \forall u : K_u^{initial} = 0, \text{ then, } MES_0 = max_u\{|K_u^{final}|\} = K_x$$

Now, if we set all of the initial estimations to zero we have:

$$\text{if } \forall u : K_u^{initial} = q, \text{ then, } MES_q = K_x + |q| \tag{2.12}$$

According to (2.12), for $q \neq 0$ we have $MES_q > MES_0$. This means that if we do not have any information about the scene, the best option for the initial values is all-zero. Any other estimations for ambiguities could cause performance loss if chosen without prior knowledge. However, if we could use the phase values to find an initial configuration such that each pixel's initial value is not more than M steps away from its optimal value, then the number of steps required to reach the ideal state will be exactly M. We propose that this reduction in the number of required steps leads to a faster execution and, therefore, will result in substantial speedup. In the next section, we verify this claim.

2.4.4 Sensitivity Analysis

In this section, we study the PrDu algorithm's run time sensitivity to variations in the initial configuration. As we mentioned in section 2.4.3, the maximum difference between the estimated and final ambiguities, i.e. MES, determines how many iterations the PrDu needs to terminate. We measure the effect of MES on the PrDu performance by running PrDu using various initial estimations on our benchmark dataset. Each configuration is a noise perturbed version of the ground truth solu-

[2]This is a safe assumption to make since the ambiguities (K) should be found relative to each other and it is possible to shift them all to satisfy this assumption. The ambiguity values are assumed to be around zero to minimize the unary terms of the energy equation as well.

tion. We control the noise magnitude to create configurations with diverse MESs. We define M as the maximum difference between initial and desired values among all variables, i.e. MES. Also, N_M represents an integer vector, with a size equal to the number of pixels, holding random values from a uniform distribution over a $[-M, M]$ range. Below is the procedure we follow to conduct this experiment:

1. We solve the phase unwrapping problem on the image with the initial configuration set to all-zero. We refer to the output ambiguities from this stage L_{GT} and the elapsed time $T_{all-zero}$. Also, we define the maximum ambiguity as $maxambig = max(abs(L_{GT}))$.

2. We build a set of initial vectors by adding L_{GT} to $N_0, N_1, ..., N_{maxambig}$ and refer to them as $G_0, G_1, ..., G_{maxambig}$

3. We apply PrDu on the same image but this time with $G_0, G_1, ..., G_{maxambig}$ as initial inputs. We let PrDu run for as many iterations as it needs until there are no moves, i.e. Up or Down, left. The output ambiguities and the elapsed times shown as $L_0, L_1, ..., L_{maxambig}$, $T_0, T_1, ..., T_{maxambig}$. Also, the speedup for each experiment calculated as $S_i = \frac{T_{all-zeros}}{T_i}$

To ensure the functionality of the algorithm after manipulating the initial inputs, we make sure that the final ambiguity outputs are the same, i.e., $L_0 = L_1 = ... = L_{maxambig} = L_{GT}$.

Fig.2.1 reports speed up and shows how changing the maximum error span, MES, affects performance. The rightmost point shows how much speedup is gained by using G_1 as the solver's initial configuration, i.e. S_1. Using G_1 means that the initial values are, at most, one step ($+1$ or -1) away from their desired, final value. Therefore, PrDu algorithm needs to run for one iteration to reach the global minima.

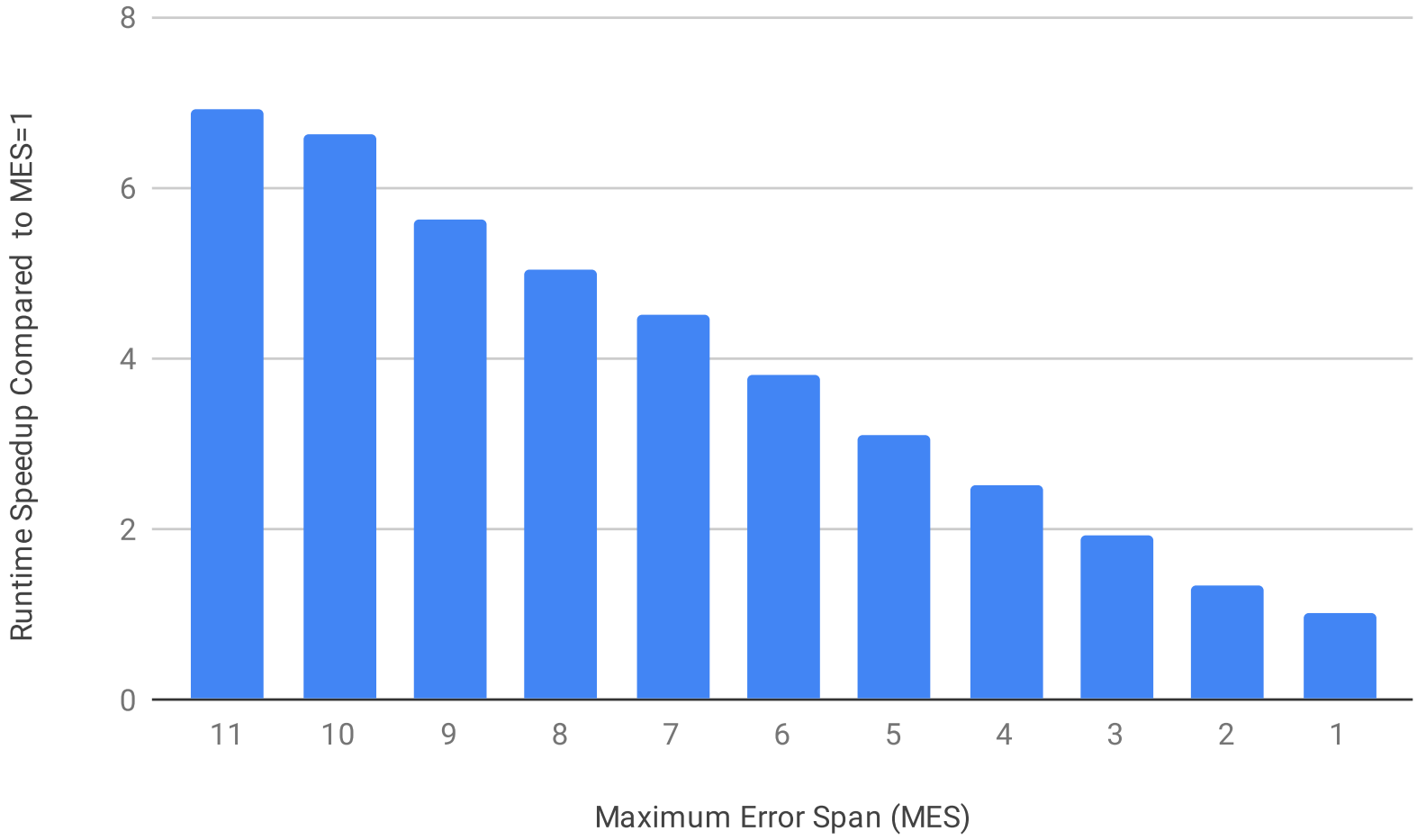


Figure 2.1: Run time speedup for different MESs. Smaller MES result in faster execution time.

By increasing the value of MES, the speedup value drops. This is due to the fact that for larger values of MES, the algorithm needs to perform more iterations to reach the final solution and thus requires more time to execute.

We conclude that the solver's run time relies heavily on the MES. This motivates us to find an initial configuration in which the MES is minimized.

2.4.5 Methodology

In the previous section, we argued that PrDu's performance could be improved significantly by providing it with a favourable initial configuration. To make use of this, we introduce the Dual-Stage method. Our goal is to divide the problem into two stages. First, we find an inexact solution for each variable (estimation) using a fast process. Second, we provide the Primal-Dual with these estimates as the initial values to obtain the exact, final solutions. The desired estimates goal is making the second stage converge fast. To gain a speedup, the run time savings of the second stage should outweigh the run time of the first stage. To obtain such estimates in the first stage, we relax the initial problem by dividing it into smaller pieces. This allows us to tackle the non-linear relationship between the PrDu time complexity and the problem size.

The Patching Method

InSAR phase unwrapping is computationally expensive on large datasets. The SAR images are typically acquired over a vast area and therefore map to an energy function with a numerous number of unary and pairwise terms, around 100 million to 1 billion of each per (modern high-resolution) interferogram. A conventional solver such as PrDu, implemented with the BK algorithm [6] as its Min-Cut/Max-Flow operation,

needs a few hours[3] to solve such problems. Moreover, at its peak, the solver will take up to tens of gigabytes of system memory over the course of its run time, which translates to a heavy memory-time requirement, which, in turn, lowers the resulting throughput of the calculation.

To overcome these stated issues, we break the problem into smaller partitions (patches). We then propose applying PrDu to solve each partition only considering the pairwise terms local to each partition. Finally, we concatenate the results together to form a global ambiguity vector. We refer to this solution as the *patching* method.

The list of pros and cons of this approach is as follows:

Pros:

1. **Higher Parallelism**: After partitioning the problem, each segment is processed separately. The tasks responsible for applying PrDu on the patches are embarrassingly parallel. Therefore, as the result of employing processing resources in our system, we run multiple instances of PrDu at the same time.

2. **Enhanced Speed**: The max-flow step of PrDu mainly dominates the time complexity of the PrDu algorithm. In practice, the BK maxflow algorithm is one of the most competitive in its class, with its run time dominated by the length of augmenting paths it computes. The time complexity in the case of the BK algorithm [6] is $O(m^2n^2log(U))$ where m is the number of edges, n is the number of variables and U is the largest capacity of the edge. By partitioning the problem into p partitions, the number of variables and pairwise terms are (roughly) divided between the partitions. Therefore, each partition now has n/p variables and m/p pairwise terms. Since the BK's run time is proportionate to

[3]On the platform specified in section 2.5.2

the square of these quantities, this will result in a quadratic reduction in the run time of each partition.

Cons:

1. **Overhead**: To achieve high performance, we need a low-overhead partitioning mechanism. Partitioning is performed only once for each stack of interferograms (since the connectivity topology, i.e. the arguments of the pairwise terms is assumed identical among the stack layers). As such, the overhead is divided among the entire stack.

2. **Inexact solution**: We dismiss the pairwise terms which have arguments from two segments. Therefore, the model presented at the first stage does not yield the optimum configuration. As a consequence, for noisy Phase Unwrapping problems, the solutions may not be close to the desired result. However, we solve this problem by feeding the output results from this method as the initial state to the global, second stage, PrDu solver[4].

We use the METIS algorithm [30] to perform partitioning. This algorithm is relatively fast and is capable of partitioning our problem so that the number of inter-segment pairwise terms are minimized[5].

[4]PrDu is guaranteed to achieve the final, global solution regardless of the initial configurations (assuming enough iterations).

[5]This is also known as edge-cut [26] and METIS can be used to directly address this objective.

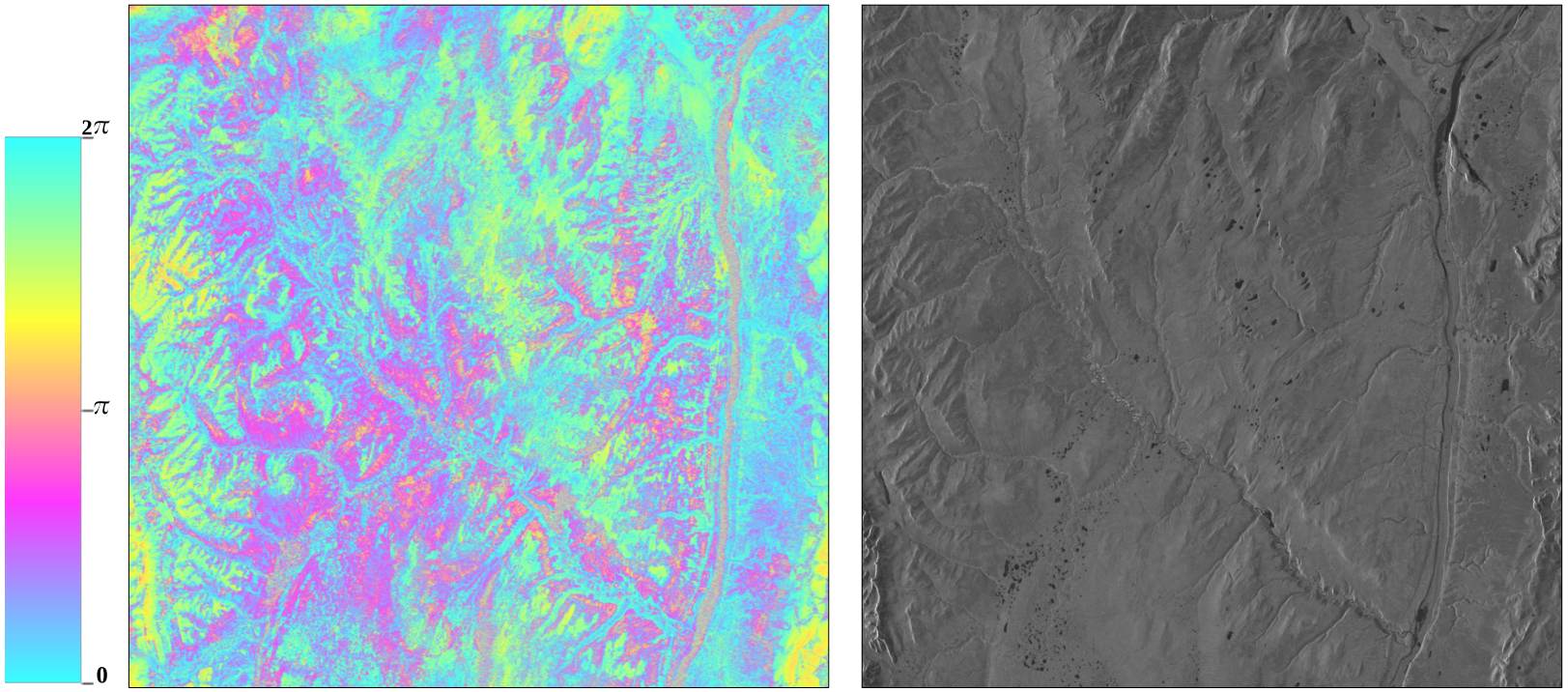


Figure 2.2: One of the 45 interferograms and amplitude scenes used as the benchmark. Phase Unwrapping is performed on the interferogram, which is the interferometric difference in phase between two SAR acquisitions over the same footprint. The high-resolution data has over 18 million data points and 100 million edges, and can be considered relatively modest in size.

2.5 Experimental Methodology

2.5.1 Benchmarks

In this section, we provide details on the InSAR taken over the Northwest Territories that we are using to assess our method's performance. We use a set of 45 interferograms (shown in Fig.2.2) to evaluate our proposed method. To generate the benchmarks, we apply a standard formulation of the Phase Unwrapping task using an energy function as described in the Problem Statement section. The corresponding energy function is convex and we then apply the PrDu algorithm as well as the Dual-Stage PrDu algorithm to the same task and present the results.

2.5.2 Measurements

We compare the proposed Dual-Stage method with the baseline method in terms of run time and memory utilization. We run the experiments on a system with Intel Core i7-8700 CPU @ 3.2 GHz with six physical cores and 32 GB of DDR4 memory with 2400 MT/s speed running CentOS Linux release 7.6. Each CPU core can run up to two simultaneous threads[6]. Therefore, in total, we can have 12 threads running concurrently. For memory utilization, we report peak resident set size (VmHWM) obtained from Linux kernel data stored under "/proc/pid[7]/status/" directory [31]. The run time is the total time it takes for the algorithm to unwrap an interferogram, in the case of the baseline PrDu solver, or a patch, in the case of the Dual-Stage PrDu solver.

[6] Using Intel Hyper-Threading technology

[7] "pid" is the process ID of the unwrapping program

2.5.3 Experimental Setup

In this section, we explain the experiments we run, and we specify the parameters used to conduct each experiment such as the number of patches and the number of parallel threads.

As our baseline, we run the conventional, single-threaded, primal-dual algorithm on the stack and measure the timing and memory utilization. To measure the effect of the patch count, we run the experiment with 16, 32, 64, 128 and 256 patches. We apply PrDu on all of the patches in a sequential manner to have a fair comparison with the baseline. Finally, we run the patching stage with 2, 4, 8 and 16 threads and report the achieved speedups.

2.6 Results and Discussion

2.6.1 Run time Breakdown

Fig.2.3 shows the run time breakdown for the patching method with a various number of patches and compares them with the baseline. We divided the run time into three parts, METIS & setup, Patches solving time and Second stage time. The METIS & data handling time stands for the time required by the METIS algorithm to partition our problem and the data handling overhead. The results show that the partitioning (patching) phase time is almost the same for the different number of patches (averaging around 18.5 seconds) and is negligible compared to the other phases. The same is applicable for the data handling time with an average of 2.7 seconds. It includes the time required to allocate new vectors and build up the patches. This accounts for 1.3% of the overall execution time and should be considered negligible.

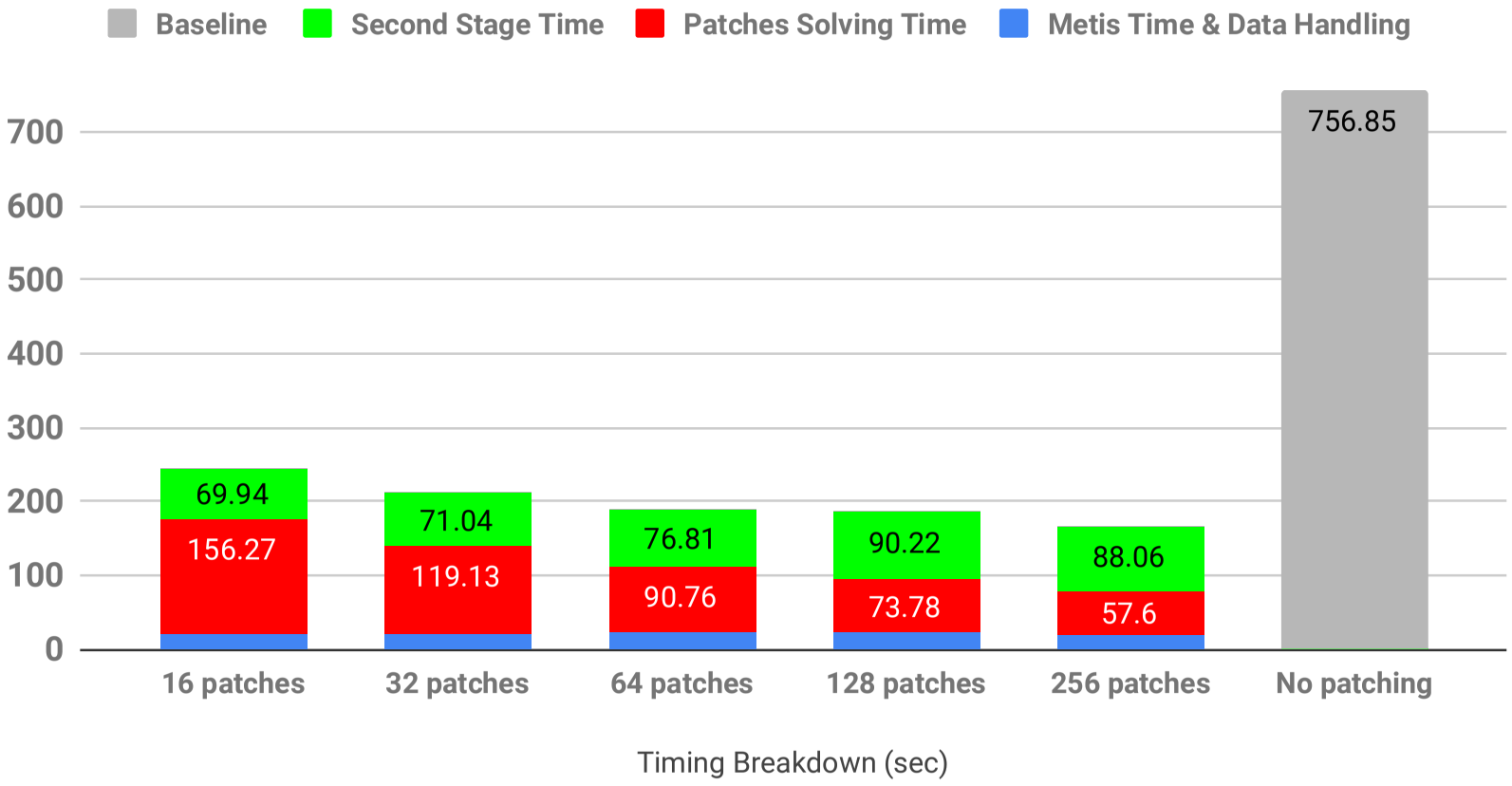


Figure 2.3: Breakdown of run time for patching method and the baseline, no patching, approach. Only single threaded results shown here.

The patches solving time includes the time required for solving the smaller problems (problem-patches) we generated in the previous step. For benchmarking purposes, we run these problem-patches serially, so the time indicates the accumulated run time of all of the patches. As can be seen in Fig.2.3, this time decreases by increasing the number of patches. The reason for this is the nonlinear relation between the problem size and PrDu run time explained in the pro's of the patching method (see section 2.4.5).

The second stage run time indicates the global PrDu solver run time after initialized by the estimated ambiguities of the previous stage. This time can be compared to the baseline time, which is the global PrDu solver. As reported in Fig.2.3, the second stage run times increase when partitioning the problem into a higher number of segments. We face a trad-off here, using larger patches to form the initialization tends to lead to more accurate estimates (result in smaller second stage time) but is more time consuming (larger patches solving time). One extreme case is a baseline which spends no time on solving the patches but spends (on average) 9.6× more time on the global solver.

2.6.2 Multi-threaded Performance

To take advantage of the parallelism introduced by the patching method, we provided the first stage with more processing cores. By doing so, we reduce the time spent on the Patches Solving phase. However, this parallelism causes an overhead which can negate the speedup beyond a certain number of threads. As reported in Fig.2.4, the improvement is not proportional to the number of processing threads. For example, in the case of 128 patches, adding four threads (from 4 threads to 8 threads) would result in 9% performance improvement while adding 8 more threads, i.e. utilizing 16

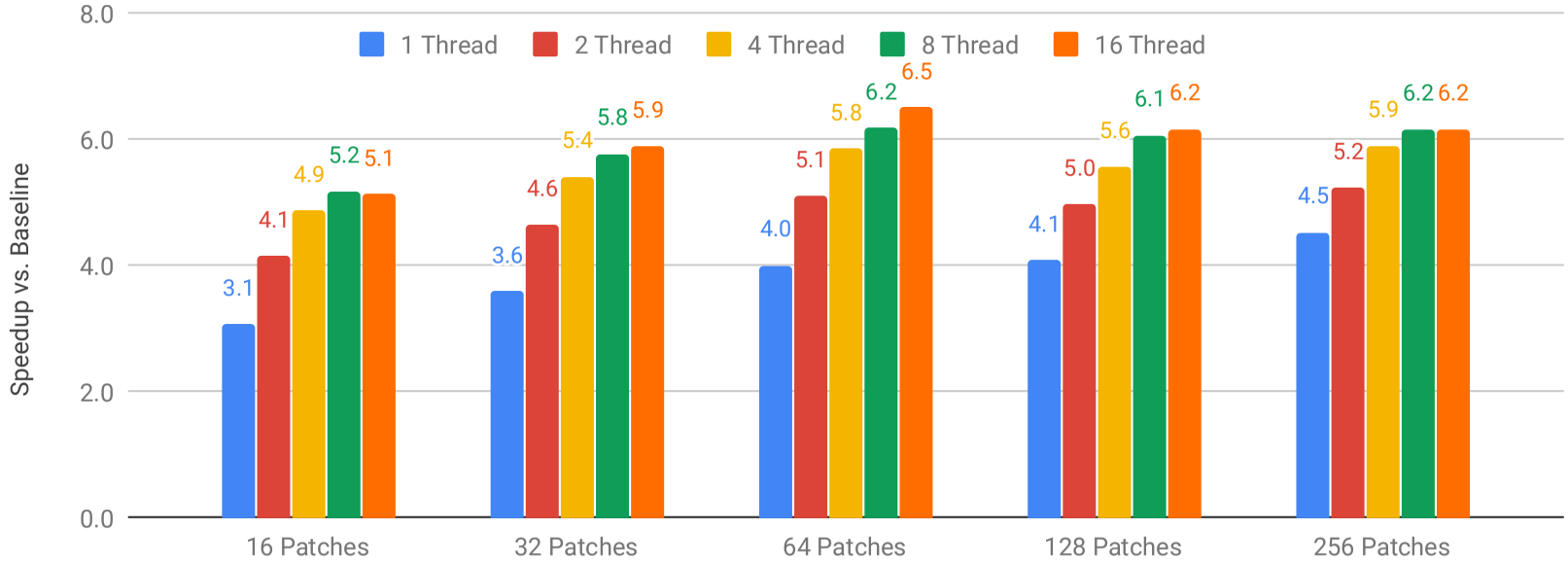


Figure 2.4: Sensitivity of the total application's speedup to the number of patches and number of processing threads.

threads, improves the performance less than 1.7%. This scenario is more dramatic for 16 patches scenario, by doubling the number of threads from 8 to 16, we end up losing performance for about 1% even after utilizing 8 more threads. These effects demonstrate the overhead impact and the importance of tuning the patching model parameters according to the data characteristics.

The fact that we are not gaining more performance (speedup) after a certain number of threads can be explained using the Amdahl's law. To do so, we need to breakdown the different portions of the algorithm as shown in figure 2.3. As explained in 2.4.5, only the first stage of the dual stage method can be executed in parallel[8]. However, other portions of the algorithm, i.e. green and blue sections in figure 2.3, should be executed in a sequential manner. We define p as the proportion of execution time that the part benefiting from parallelism. In our case:

$$p = \frac{\text{Patches Solving Time}}{\text{Total Runtime}}$$

$$\text{For 64 patches: } p_{64} = \frac{90.76}{76.81 + 21.86} = 0.9198$$

According to Amdahl's law, assuming that we can get s times speedup for the parallel portion (ideally by using s threads), the overall speedup of the system, S_t, can be found as:

$$S_t = \frac{1}{(1-p) + \frac{p}{s}} = \frac{s}{0.0802 * s + 0.9198}$$

So, theoretically, having two threads to run the 64 patches should give us 1.85 times

[8]There are no dependencies between the expectations of the patches, and thus they are completely independent of each other. This type of parallelism is also known as Ëmbarrassingly Parallel¨ff

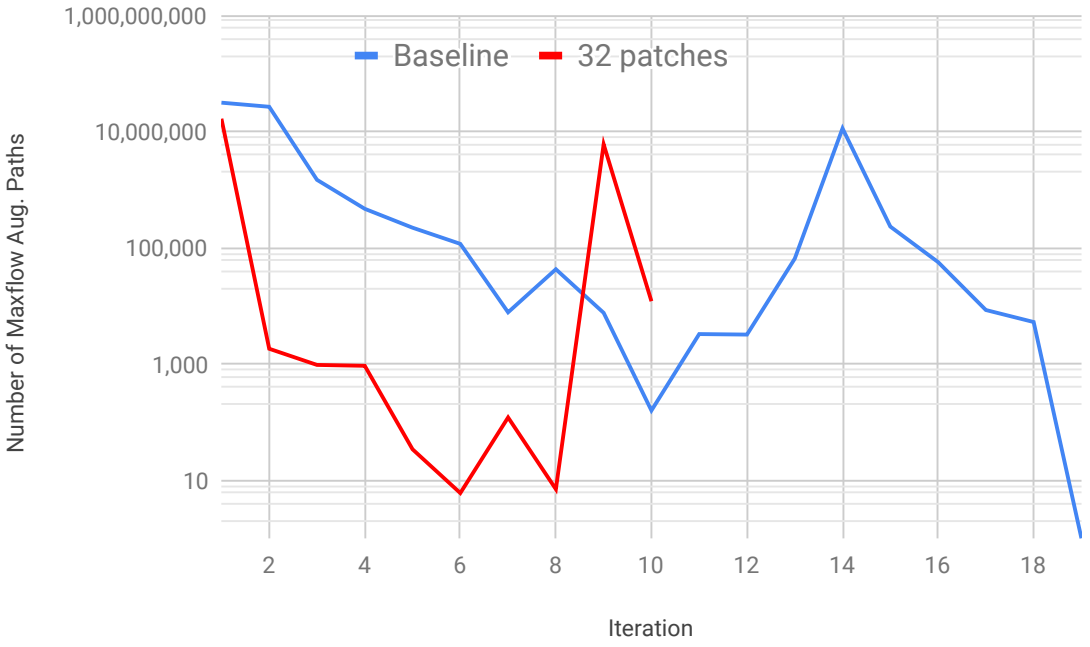
improvement as compared to the single thread performance. The single thread performance is 4.0 times faster than the baseline and thus we should expect $4.0 * 1.85 = 7.4$ times speedup. However, we get only 5.1 times speedup ($30\%lower$) due to other overheads in running the parallel threads. The performance drop (compared to the ideal scenario) is even more drastic when we increase the number of threads to 16. In that case, $S_t = 7.26$ which translates to $4.0 * 7.26 = 29$ times speedup compared to the baseline. However, in practice we can only get 6.5 times improvement which is a drastic 75% lower than ideal.

2.6.3 Memory Overhead

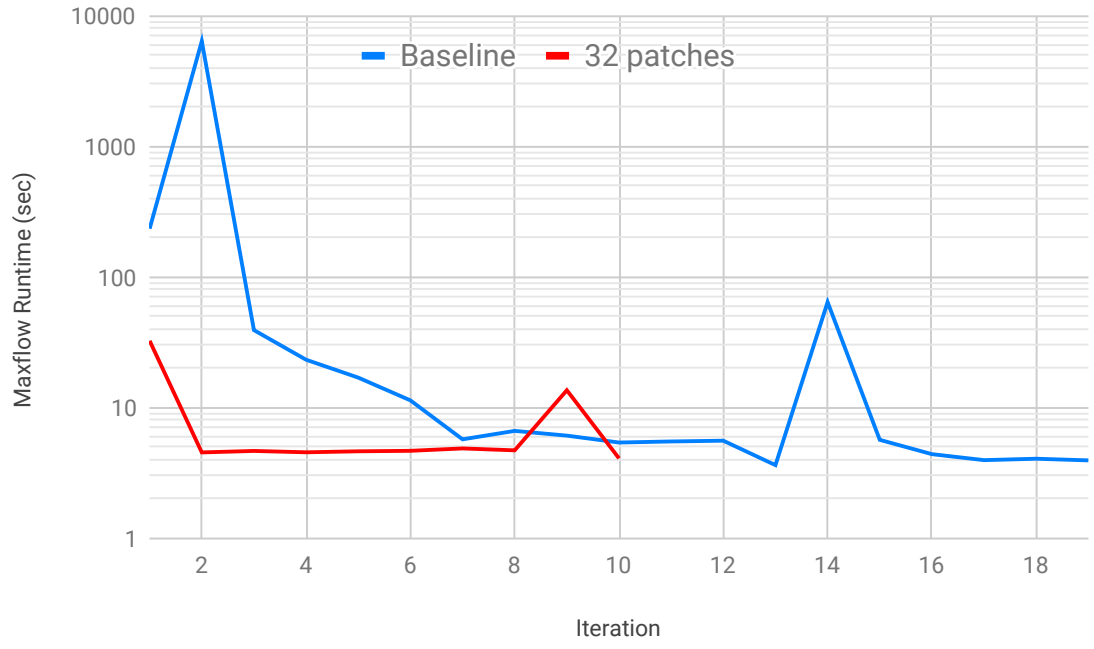
It is worth mentioning that employing the Dual-Stage method has a memory usage overhead. Our method utilizes 21.4% more main memory compared to the baseline. A part of this overhead is because of the METIS algorithm and the rest is used to manage the partitioning and concatenation logic. The memory overhead is constant among different configurations (various number of threads and patches). The reason for this consistency is that the size of the data structures that store the graph, points and partitions do not change from one configuration to another.

2.6.4 Explaining the Speedup

As we saw in section 2.6.1, the speedup obtained by Dual-Stage algorithm is not linearly correlated to the number of threads or the number of patches. For example, using 16 patches and a single thread results in more than three times performance improvement as compared to the baseline. To investigate the reason behind this non-linear speedup, we investigated the part of the algorithm that dominates the run time: the maxflow computation. For the last stage of the baseline solver, the max-flow computation takes 99.5% of total solver time, and 87%, on average over all



(a)



(b)

Figure 2.5: a)Number of Augmenting Paths, i.e. length given in number of edges traversed) and b) Run time for different maxflow iterations. As can be seen the baseline needs more iterations to converge. Blue line represents the baseline PrDu and the red line indicates the quantities for the global, second stage, PrDu.

configurations. The run time of the maxflow computation using the BK algorithm is determined by the number and the length of augmenting paths in each iteration. Empirically, the BK maxflow algorithm is most competitive when the longest paths it augments are short. [16].

Fig.2.6.3 shows the number of augmenting paths of all maxflow calculations of the global PrDu algorithm. Global PrDu runs on both baseline and Dual-Stage configurations (second stage). As can be seen, the patching technique significantly reduces the number and length of augmenting paths and also the number of required iterations compared to the baseline. The latter can be explained by initialization with a lower MES. We hypothesize that the number and length of augmenting paths is greatly reduced in the Dual-Stage approach, and hence the observed speedup of the maxflow computations in the Dual-Stage PrDu. Evidence of this can be seen by comparing the two figures in Fig.2.5. These figures show a correlation between the number of augmenting paths and the run time, and also a drop in both these quantities between the baseline and Dual-Stage approach. Moreover, a 40x drop is observable in the 90^{th} length of the augmenting paths between the first 10 iterations of baseline and Dual-Stage approach.

2.7 Conclusion

An efficient method for accelerating large scale convex energy minimization is presented. The method breaks the problem into patches to obtain a sub-optimal initialization, then uses the PrDu algorithm to reach the global optimum. We also introduced InSAR Phase Unwrapping as an application which benefits from this method. We evaluated our method on InSAR images taken by the Radarsat 2 sensor and gained up to 6.5 times speedup. Moreover, we showed the impact of initialization

on PrDu performance. In the end, we investigate the maxflow operations within the PrDu algorithm to justify the speedup.

Chapter 3

RAB: A Receiver-Aware Bandwidth Saving Solution

3.1 Abstract

The past few years witnessed a rise of demand for online video streaming. This rise in demand leaves the streaming companies with a large number of customers to service and thus a significant amount of bandwidth to pay for. Although the streamers use compression and encoding techniques, the clients' quality of experience prevents them from overplaying these techniques. Simultaneously, there has been a growing interest in another venture of research that focuses on developing techniques that aim to restore the quality of the videos that have been subject to compression. The presence of such techniques is often ignored by the transmitter. In this paper, we propose Receiver-Aware Bandwith Saving Solution (RAB). RAB aims to reduce the utilized bitrate by taking advantage of the benefits of having some enhancement techniques on the receiver's side. We prove our concept by employing a particular enhancement

technique, and show the ability of our technique to reduce utilized bandwidth without compromising the client's quality of experience.

3.2 Introduction

With the rise of technology, TVs with higher resolutions have become popular. Moreover, the majority of the video contents are digitally streamed and are transmitted over the internet. According to a report by Sandvine [14], in 2019 more than 55.4% of the global traffic share was utilized by video streaming. This rate had risen by 2.2% as a result of the ongoing pandemic[14].

Costs of operations for streaming companies such as Netflix are highly dominated by cloud hosting and content delivery. According to Intricately [27] estimates, Netflix spent more than 420 million dollars on acquiring bandwidth in 2018. Despite efficient compression and encoding techniques, the economic burden of utilizing such huge bandwidths still looms over the financial statement of the streaming companies. Therefore, any bandwidth savings, while maintaining the quality of experience (QoE), will benefit the industry.

There are two ways to reduce the required bandwidth. The first is to reduce the number of pixels in each frame. The second is to manipulate the compressor/encoder parameters so that the resulting bitstream contains less data per unit of time. Previous work has investigated the optimal resolution for any given bandwidth. Such solutions can be simple, e.g. using a fixed lookup table or very complicated, e.g. per-title encode optimization. In [39], the authors examined different variations of these adaptive bitrates (ABR) systems and categorized them according to their criteria, nature and adaptation module. However, previous works often ignore the capabilities

of the receiver and post-decoding apparatus. In this work, we address this inefficiency by taking into account the receiver resources as well as the channel's limitation to decide the transmission bitrate. To this end, we assume that an "Initial Bitrate" is selected for the transmission. We then modify this bitrate according to the receiver's capabilities. By doing so, we reduce bitrate without compromising QoE.

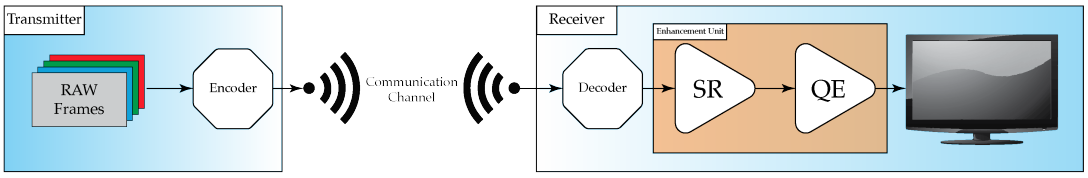


Figure 3.1: Video transmission pipeline. The client (receiver) is equipped with some enhancement techniques (SR and QE). The positions of SR and QE may be different.

In recent years, there is a growing interest in another venture of research that focuses on developing systems that aim to restore the quality of the videos that have been subject to compression or down-sampling using the following approaches:

- "Quality Enhancement" (QE): which removes the compression artifacts introduced by the encoding process.

- "Super-Resolution" (SR): which increases resolution by restoring the information lost during the down-sampling process.

Figure 3.1, shows the transmission pipeline equipped with the enhancement techniques. The streamer holds the RAW frames in its database which is encoded before travelling through the communication channel. After traversing the network, the decoder turns the bitstream into video frames. These frames pass through QE and/or SR methods (whichever applicable) before being displayed on the user's screen.

The presence of QE and SR is often ignored by the transmitter. In this paper, we propose Receiver-Aware Bandwith Reduction (RAB). RAB aims to reduce the utilized

bitrate by taking advantage of the benefits of having some enhancement techniques on the receiver's side. In the interest of space we focus on a particular QE; leaving the SR-based solutions to future studies.

As presented in Figure 3.1, without RAB, the encoder encodes the RAW frames to the point that the resulting bitstream's bandwidth does not exceed the communication channel's limitations (Initial Bitrate). However, RAB examines the frames, the client's QE, and the initial bitrate to decide on how further the encoder is allowed to compress the frames. The main criteria limiting the extent of the compression is the client's QoE: the client should not see a difference between these two scenarios:

- No QE, No RAB
- With QE, With RAB

Note that the latter utilizes less bandwidth which will benefit the stream provider.

The rest of the paper is organized as follows. In the next section, we discuss the background and methodology. Section 3.4 explains the RAB in detail. Section 3.5 reports our experimental methodology. Section 3.6 includes results, and section 3.7 concludes the paper.

3.3 Background & Related Works

3.3.1 Encoding artifacts

One way to evaluate the encoding process is to study the Rate-Distortion (a.k.a. R-D) curve. Figure 3.2 presents bitrate (Rate) vs. quality (i.e. inversely proportional to Distortion) for a typical encoder (like HEVC or AVC). As presented the slope of the

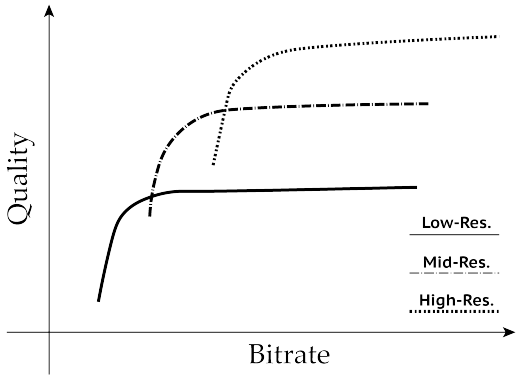


Figure 3.2: Relation between perceived video quality and target bitrate for different resolutions of typical video content.

curve steepens when moving left (lower bitrate). This is explained by the nature of encoding algorithms. Both encoders produce compressed streams by quantizing the coefficients of Discrete Cosine Transform (DCT). Various bitrate targets are achieved by changing the step size of this quantization. Quantizing the coefficients is a lossy procedure and a bigger step size (aggressive encoding) can reduce quality significantly. These compression artifacts are referred as "blocking", "ringing" or "mosquito noise". In such cases, the video appears "noisy" or "low quality". In section 3.3.2 we discuss two methods to quantify the quality of a video.

3.3.2 Quality Assessment — PSNR & VMAF

One of the most challenging tasks in developing an image/video -based system is to assess the quality of the resulting image, a.k.a. Image Quality Assessment (IQA) [43]. This is because there are two fundamentally different perspectives in measuring the quality of any signal (including images): First is the subjective perspective. Subjective evaluation requires a human being to evaluate the image in question. Since subjective evaluation uses humans (and is hence expensive and time consuming), the results are expected to be highly accurate. On the other hand, using an objective

IQA would be faster, cheaper and more accessible. IQAs can be mathematical, e.g. Peak signal-to-noise ratio (PSNR), or based on the biology/psychology of humans, i.e. Human Vision System (HVS) -based metrics. In this paper, we use two objective metrics, PSNR and Video Multi-Method Assessment Fusion (VMAF) [34].

Although PSNR has become the de facto standard for quality measurement, it is not always consistent with humans' perceived quality. To overcome this flaw, Netflix introduced the VMAF [34] quality metric. VMAF predicts the perceptual quality by feeding multiple quantitative metrics (such as PSNR, SSIM, ...) into a machine learning (ML) model which then "fuses" them into a single number. The ML model is then trained using the subjective scores given by humans. VMAF enables us to estimate a visual subjective score without compromising the convenience of a quantitative approach.

3.3.3 Quality Enhancement — MFQE

In recent years, with the rise of demand for online video streaming over the limited-bandwidth internet, video compression has become increasingly popular. However, the more compressed the videos are, the lower the QoE becomes for the final viewer. To address this issue, many studies have explored removing these compression artifacts [15, 45, 44, 29, 25]. In this paper, and in the interest of space, we focus on Multi-frame Quality Enhancement for Compressed Video, MFQE v2.0 [24], as the quality enhancement unit on the decoder side.

Unlike the other approaches that use a single frame, MFQE takes into account the similarities between consecutive frames. R. Yang, et al. investigate the quality fluctuations among the encoded frames and locate the ones with higher SNR, a.k.a. Peak Quality Frames (PQFs). They utilize a convolutional neural network (CNN) to en-

hance the non-PQF frames. The compressed frame and the neighboring PQFs are used as the inputs to CNN to generate the enhanced frame. The resulting enhanced video sequence achieves higher quality and fewer quality fluctuations when compared to other state-of-the-art approaches.

3.4 RAB

RAB aims to split the advantages of having a video enhancement unit between the streamer and the viewer. Conventionally, Quality Enhancement (QE) units improve the Quality of Experience (QoE) of the end-user. Their technical benefits for the providers, i.e. the streaming companies, are insignificant. The RAB allows the streamers to benefit from QEs by reducing the required bandwidth. Figure 3.3 presents a high-level perspective of the streaming pipeline, with and without the RAB and the QE.

As presented in Figure 3.3.c., RAB sits right behind the encoder on the transmitter side. It uses the information from the receiver side (labelled as metadata) to tune the encoder's parameters. This information may include network and other system components' specific details such as display size, GPU or battery life [39]. In this work, we assume that the client may or may not have a QE unit receiving the decoded video frames (Figure 3.3.b.). We also assume that the RAB is aware of the existence of such a unit if in fact it is provided. Depending on whether or not the RAB and/or the QE exists, three scenarios are possible. The first scenario assumes the baseline model, i.e. there is no QE on the client's side, and the streamer is not equipped with the RAB. Under this scenario, the encoder considers only the network limitations for its encoding process. On the receiver side, the video frames arrive at the user's display without any modifications after being decoded. We present this scenario in

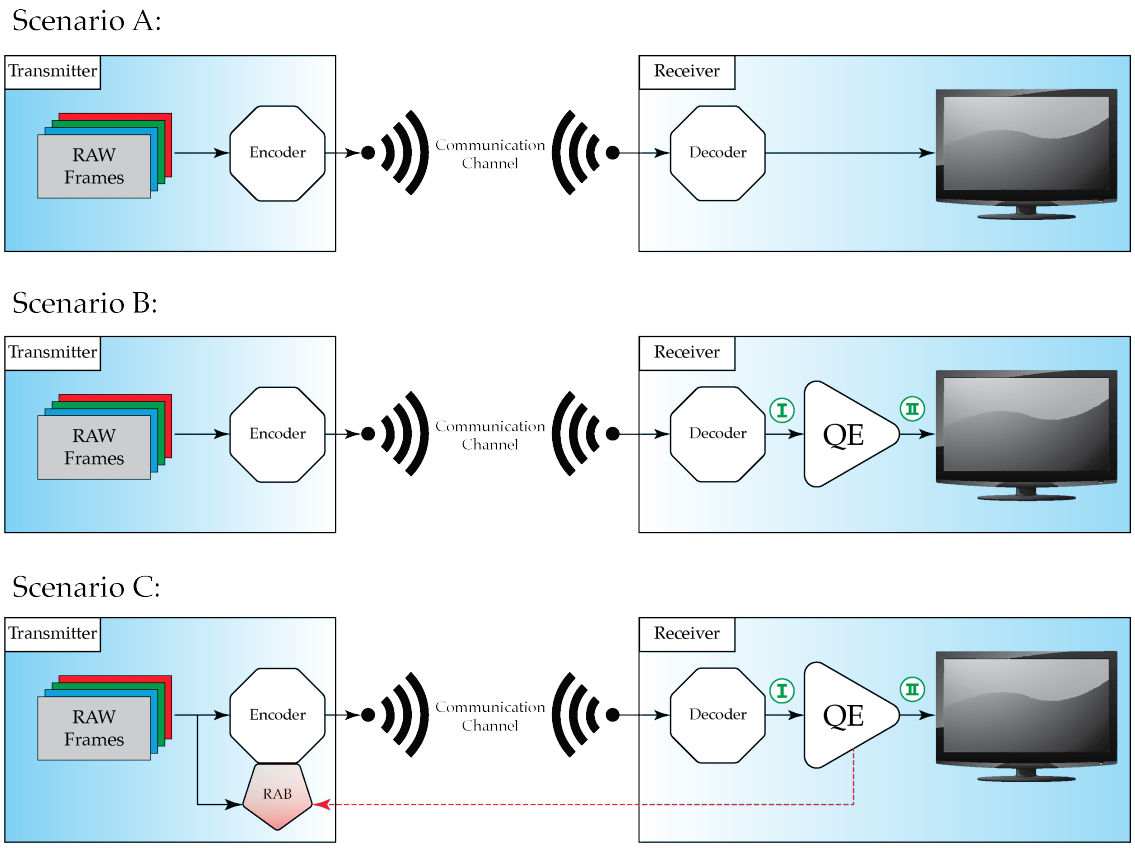


Figure 3.3: Three possible setups for utilizing QE and the RAB: A. Conventional setup, without the RAB and QE. B. The client has a QE to improve her image quality. C. The client has a QE and the streamer installed the RAB to save on bandwidth. The quality loss will be compensated for by the QE.

Figure3.3.a.

The second scenario assumes that the RAB does not exist, but the QE is available on the client's side (3.3.b.). Here the decoded frames pass through a QE unit which, compared to the first s improves the user's experience quality. However, the transmitter's experience and the bandwidth used stays unaffected. Therefore, from the stream provider's point of view, the cost of transmission does not change.

Our solution builds on the third scenario. RAB's allows the provider to take advantage of having a QE unit on the client's side by compressing the video to the extent where

the QE has the ability to restore the lost quality. To manage the compression rate of the encoder, we look for a "Target Bandwidth" by taking into account the strength of the QE, and the "Initial Bandwidth" provided by the network. Below we explain our assumptions and approach.

3.4.1 Assumptions

We assume that there is a standard, universal QE unit. While this does not hold under the current state of the technology, we assume that in the near future, along with new encoding standards, it will. We also assume that in the foreseeable future there will be a standard way of improving, i.e. denoising or increasing the resolution of the encoded videos after they are transmitted over to the receiver[1].

We also assume that the transmitter is equipped (or is fully aware of) with the aforementioned standard QE and can use it in real-time or in an offline manner. This capability will help the RAB decide about how intensely it can manipulate the encoder's parameters. For example, if a certain video is unlikely to improve by the QE, then no benefit can be extracted by reducing the bandwidth without compromising the viewer's quality of experience. On the other hand, if the QE is able to improve the entire video content consistently, steady bandwidth reduction can be achieved.

3.4.2 Method

In this section, we describe how the RAB works. The algorithm takes three inputs:

I) Information regarding the existence of a QE unit.

II) The Initial Bitrate (BR), i.e. the bitrate that would be the target bitrate of

[1]This method can be based on theoretical analysis, like [35], or Machine Learning (ML) -based, like the QE that we are using in this paper, [24].

the encoder if the RAB is not used. This bitrate is usually decided by the capabilities of the network.

III) The R-D curve or an estimation of it is provided as a function. This function, $F_{RD}(.)$, derives the quality (PSNR[2]) as a function of bitrate for any particular content and encoder. This function can be calculated offline regardless of the QE.

Assuming that these three inputs are provided, following the steps below will generate the Target Bitrate:

1. Encode and Decode the video using the Initial bitrate. Then, apply QE on the decoded video and generate an enhanced version

2. Find the enhance factor (EF)
   $(EF = \text{PSNR}^{Enhanced} - \text{PSNR}^{Reconstructed})$

3. Calculate the Bitrate Saving amount, α:
   $\alpha = EF * (\partial F_{RD}/\partial \text{Bitrate})$

4. Calculate the target bitrate:
   Target BR = Initial BR $- \alpha$

There are a few points to consider when using the proposed method:

- Steps 1 through 3 can be calculated offline and kept in a look-up table. This table then can be used by step 4 in real-time. Note that such a look-up table is only valid for a pair of video sequence and a QE unit.

[2]Note that we are not using VMAF in our algorithm. We only use VMAF to evaluate the final result

- The enhance factor calculated in step 2 is a function of the video content and the initial bitrate for which the video was encoded for.

- The function that calculates α from EF, used in step 4, uses the slope of the R-D curve ($\partial F_{RD}/\partial$Bitrate). The value of α determines how further left we can go, i.e. how much we can lower the bitrate, without losses exceeding EF from the PSNR. More details can be seen in Figure 3.4. Note that we rely on a linear estimation of the R-D curve. Also, we assume that EF stays the same in the proximity of the initial bitrate (noQE) point. These assumptions hold as long as α is small.

- For simplicity, a fixed value can be used instead of $\partial F_{RD}/\partial$Bitrate as long as the change in Bitrate, i.e. α is small. To find a valuable estimation of this value in a give bitrate, B, we found the PSNR for the encoded sequences in BR= B and BR= $0.9 * B$, and divided their difference by $0.1 * B$. This gives us a rough estimation of the R-D curve slope in BR=B.

3.5 Experimental Methodology

To evaluate our solution we evaluated RAB using multiple video sequences taken from the Join Collaborative Team on Video Coding (JCT-VC) dataset[4]. As mentioned earlier, we use MFQE v2.0[24] as our Quality Enhancement (QE) unit. To demonstrate the effect of the video content on bandwidth savings, we selected four videos of that the JCT dataset, and averaged all of the measurements. To quantify the quality measures, we used the conventional PSNR, and VMAF, that is more in line with the human's perceived quality of the video sequences.

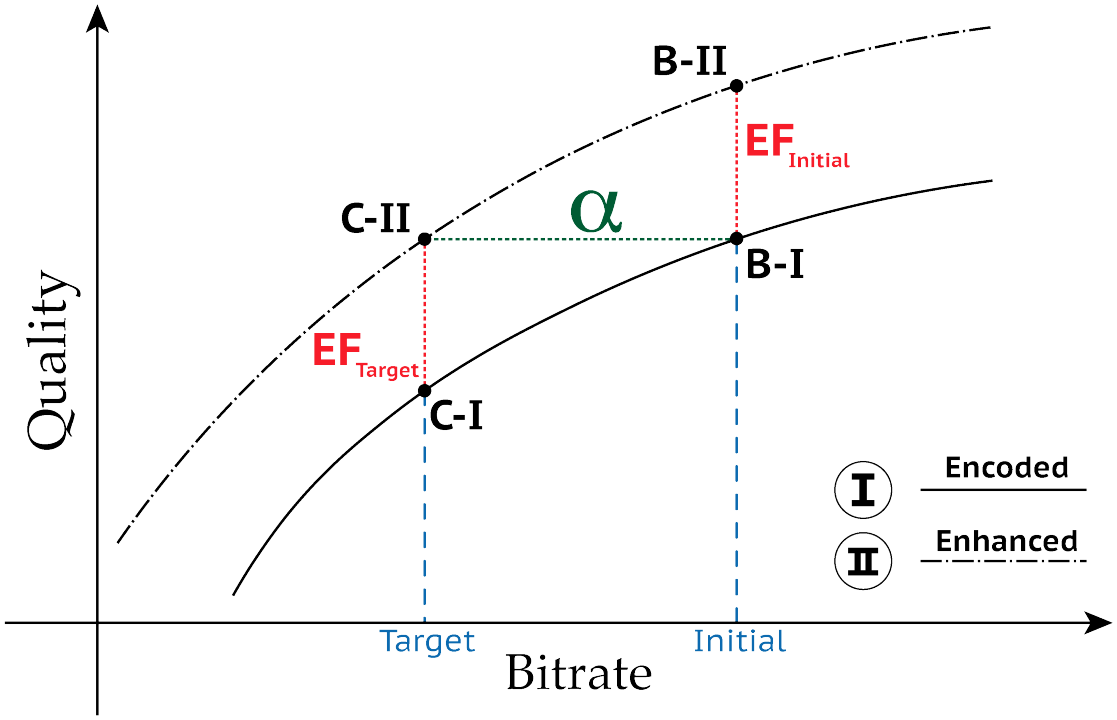
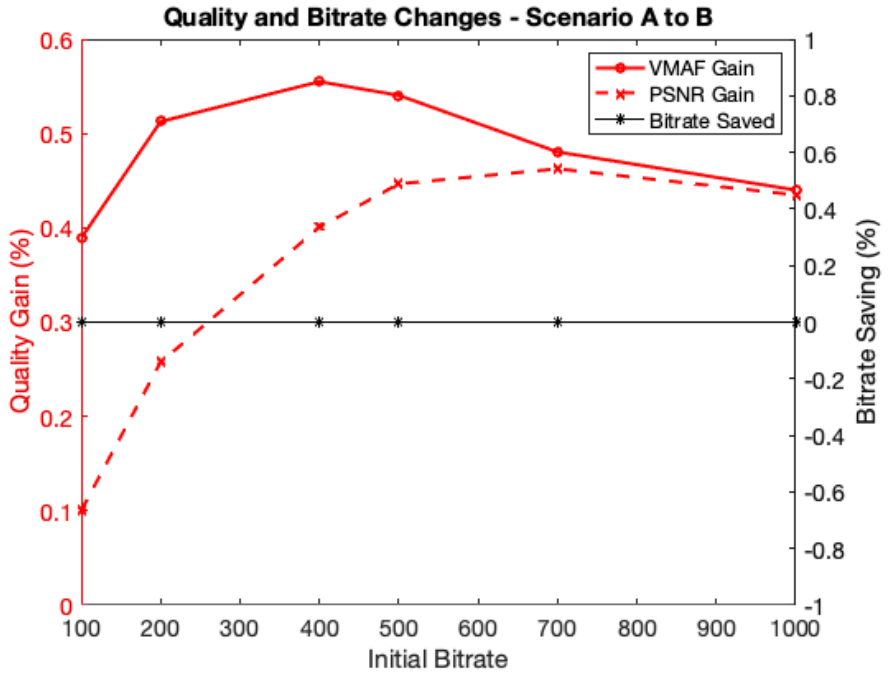


Figure 3.4: The R-D curve for decoded videos before and after applying QE, labeled as Encoded and Enhanced respectively. Point's labels are referring to Figure 3.3.

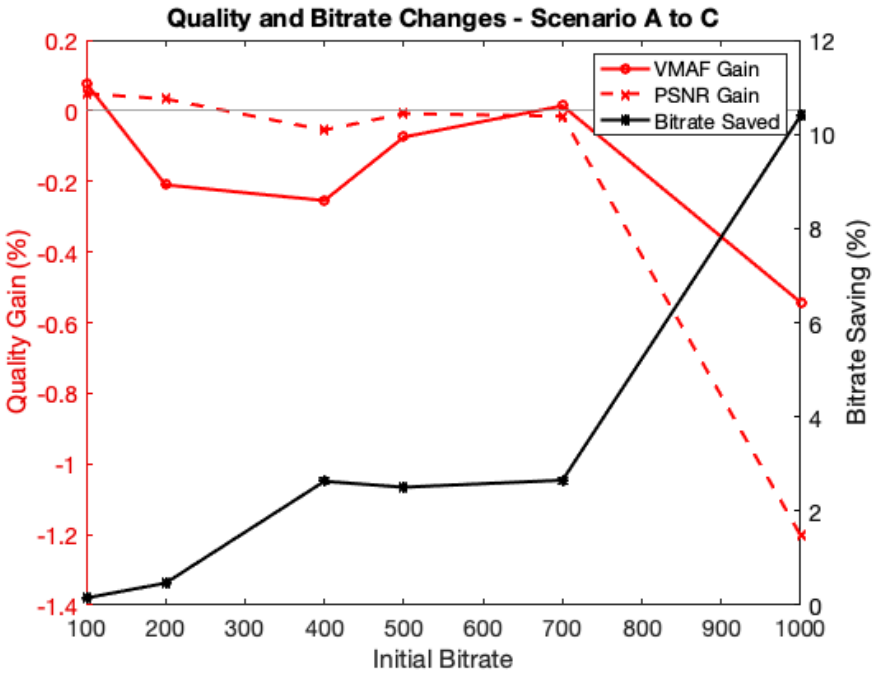
3.6 Results & Discussion

Figure 3.6 shows the quality and bitrate (BR) measurements for different scenarios presented in Figure 3.3 and detailed in section 3.4.2. Figure 3.6.a compares the scenario A, the baseline, and scenario B where QE exists but the transmitter is unaware of that. Figure 3.6.b compare scenario A and C, where RAB comes. The horizontal axis is showing the Initial Bitrate and the left vertical axis shows the amount of BR savings achieved. The right vertical axis (shown in red) shows the changes in quality (measured in terms of VMAF and PSNR shown in solid and dashed lines respectively).

As shown in Figure 3.6.a, although there is a small gain (under 1%) in quality because of the existence of the QE, there is no bitrate saving achieved. Recall that in scenario

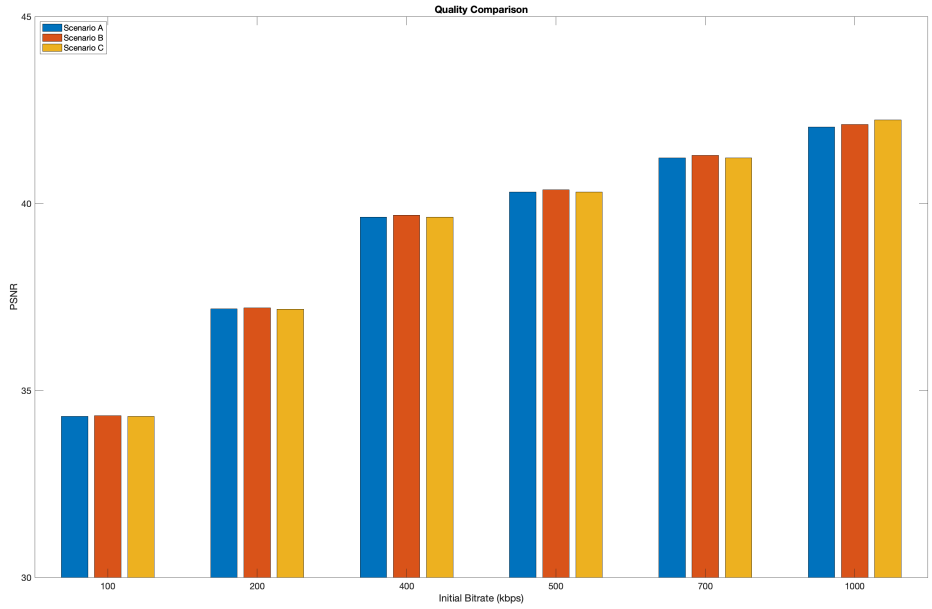


(a) Scenario B compared to scenario A

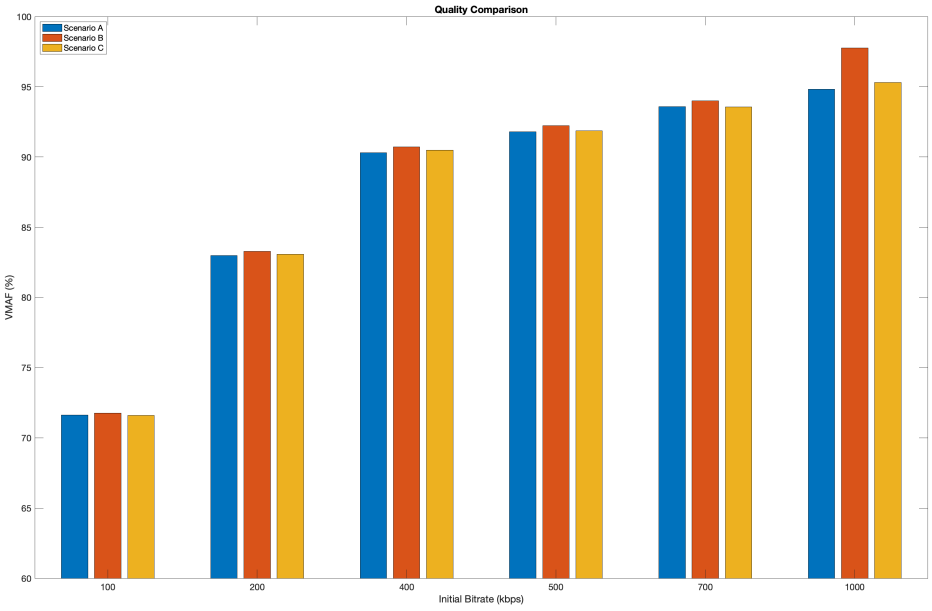


(b) Scenario C compared to scenario A

Figure 3.5: Comparing Quality and Bitrate saving metrics between different scenarios.



(a) PSNR measurements for three scenarios



(b) VMAF measurements for three scenarios

Figure 3.6: Comparing absolute quality between different scenarios in terms of VMAF and PSNR.

B the transmitter is unaware of the capabilities of the receiver and thus does not touch the conventional transmission process.

In scenario C (Figure 3.6.b), using the algorithm explained in section 3.4.2, a new target bitrate is calculated which allows the streamer to save up to 10% of the allocated bitrate (showed by the black line). However, increasing the amount of compression by RAB will reduce the quality of the videos received by the decoder. In this scenario, the QE on the client side will revert this loss in quality, and ideally generates the same perceived quality as the scenario A.

There are a couple of factors that affect the amount of achieved bitrate saving. First is the QE's enhancement capacities. A better QE technique and/or an easier-to-enhance content will increase the value of EF that is directly proportional to α, i.e. the BR saving. Secondly, with a fixed EF, the slope of the R-D curve ($\partial F_{RD}/\partial$PSNR) determines how conservative the RAB should be in reducing the target bitrate. For example, the slope is lower in higher BR (see Figure 3.2). Therefore, RAB ca reduce the BR significantly knowing that the reduction in quality is relatively small and can be compensated for by the QE. This phenomenon explains the rise of the BR savings in the higher initial bitrates.

It is worth noting that using RAB is not particularly beneficial the end-user consumer and therefore will not make any financial sense for them to upgrade to a system (TV) equipped with a QE unit while their transmitter is using RAB. However, this framework targets the streamers which also manufacture their receivers, such as Apple TV from Apple and Fire TV by Amazon. Such services can benefit from RAB framework, and easily justify the higher initial cost of the hardware by considering the lower overall bandwidth cost over the years.

3.7 Conclusion

In this paper, we introduced the RAB, an adaptive bitrate algorithm that takes into account the existence of a quality enhancement unit on the receiver side. The goal was to share the benefits of such a unit between the end-user and the streaming service provider. This goal is achieved by reducing the target bitrate of the encoder in the presence of a QE unit. We demonstrated the abilities of our solution in bandwidth reduction, without compromising the viewers Quality of Experience (QoE).

Chapter 4

Conclusions

In this work, we investigated two image and video processing applications, Phase Unwrapping (PU) in chapter 2 and Video Encoding in chapter 3. We proposed a couple of approaches to improve the performance of the former and reduce the cost of the later without compromising the outcome for neither of the applications.

In chapter 2 we summarized Interferometric Synthetic Aperture Radar (InSAR) imagery and explained the necessity of employing PU methods (chapter 2.3.2) for analyzing such images. Before explaining the Dual-Stage method in section, an overview of the existing PU algorithm was discussed in section 2.3.3. The proposed method breaks one of the PU algorithms, Primal-dual, into two stages (as the name, Dual-Stage, suggests). In the first stage, the problem is broken into several tiles, each of which can be solved independently. The second stage of the algorithm runs on the whole image but uses the results of the prior stage as an initial condition. This initial condition drastically reduced the cost and improved the timing of the commonly used Primal-dual (PD) algorithm. We provided a sensitivity analysis of the PD algorithm

measured explicitly for the InSAR application. Then we expanded the improvement results of our algorithm and analyzed its different aspects, from memory and time breakdowns to multi-threaded performance comparison.

In chapter 3 we focus on a completely different problem, video streaming. We started by explaining the video streaming system as a whole, from the streamer to the end-user. Also, we pointed out another venture of research that aims to restore the quality of the videos lost during the encoding process, a.k.a. Quality Enhancement (QE). Then we introduced our approach, RAB: A Receiver-Aware Bandwidth Saving solution, along with a couple of quality measures to assess and quantify the improvements. We showed that if the transmitter is aware of the existence of a QE unit on the receiver side, it can use this knowledge to change how it encodes and sends the video to the receiver. Finally, we set up a set of experiments to prove our concept and to show that it is possible to achieve a considerably lower bandwidth utilization without compromising the user's quality of experience.

Bibliography

[1] Jos M Bioucas-Dias and Gonalo Valadao. Phase unwrapping via graph cuts. *IEEE Transactions on Image processing*, 16(3):698–709, 2007.

[2] José M. Bioucas-Dias and José M. N. Leitão. The zm algorithm: a method for interferometric image reconstruction in sar/sas. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 11 4:408–22, 2002.

[3] José M Bioucas-Dias and Gonçalo Valadão. Discontinuity preserving phase unwrapping using graph cuts. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 268–284. Springer, 2005.

[4] Frank Bossen et al. Common test conditions and software reference configurations. *JCTVC-L1100*, 12:7, 2013.

[5] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

[6] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions*

on Pattern Analysis & Machine Intelligence, (9):1124–1137, 2004.

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 377–384. IEEE, 1999.

[8] J. R. Buckland, J. M. Huntley, and S. R. E. Turner. Unwrapping noisy phase maps by use of a minimum-cost-matching algorithm. *Appl. Opt.*, 34(23):5100–5108, Aug 1995.

[9] C. Cafforio, C. Prati, and F. Rocca. Sar data focusing using seismic migration techniques. *IEEE Transactions on Aerospace and Electronic Systems*, 27(2):194–207, March 1991.

[10] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, May 2011.

[11] C. W. Chen and H. A. Zebker. Phase unwrapping for large sar interferograms: statistical segmentation and generalized network models. *IEEE Transactions on Geoscience and Remote Sensing*, 40(8):1709–1719, Aug 2002.

[12] M Costantini and Paul A Rosen. A generalized phase unwrapping approach for sparse data. In *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No. 99CH36293)*, volume 1, pages 267–269. IEEE, 1999.

[13] Mario Costantini, Fabio Malvarosa, and Federico Minati. A general formulation for redundant integration of finite differences and phase unwrapping on a

sparse multidimensional domain. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):758–768, 2011.

[14] Cam Cullen. Covid internet phenomena spotlight report. Available at https://www.sandvine.com/covid-internet-spotlight-report (2021/01/14).

[15] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 576–584, 2015.

[16] Barak Fishbain, Dorit S Hochbaum, and Stefan Mueller. A competitive study of the pseudoflow algorithm for the minimum s–t cut problem in vision applications. *Journal of Real-Time Image Processing*, 11(3):589–609, 2016.

[17] Thomas J Flynn. Two-dimensional phase unwrapping with minimum weighted discontinuity. *JOSA A*, 14(10):2692–2701, 1997.

[18] Gianfranco Fornaro, Giorgio Franceschetti, Riccardo Lanari, Eugenio Sansosti, and Manlio Tesauro. Global and local phase-unwrapping techniques: a comparison. *JOSA A*, 14(10):2702–2708, 1997.

[19] Andrew K Gabriel, Richard M Goldstein, and Howard A Zebker. Mapping small elevation changes over large areas: differential radar interferometry. *Journal of Geophysical Research: Solid Earth*, 94(B7):9183–9191, 1989.

[20] Dirk Geudtner, Ramón Torres, Paul Snoeij, Malcolm Davidson, and Björn Rommen. Sentinel-1 system capabilities and applications. In *2014 IEEE Geoscience and Remote Sensing Symposium*, pages 1457–1460. IEEE, 2014.

[21] Dennis C. Ghiglia and Louis A. Romero. Robust two-dimensional weighted and

unweighted phase unwrapping that uses fast transforms and iterative methods. *J. Opt. Soc. Am. A*, 11(1):107–117, Jan 1994.

[22] Dennis C Ghiglia and Louis A Romero. Minimum lp-norm two-dimensional phase unwrapping. *JOSA A*, 13(10):1999–2013, 1996.

[23] R. M. Goldstein, H. A. Zebker, and C. L. Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science*, 23(4):713–720, July 1988.

[24] Zhenyu Guan, Qunliang Xing, Mai Xu, Ren Yang, Tie Liu, and Zulin Wang. Mfqe 2.0: A new approach for multi-frame quality enhancement on compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[25] Jun Guo and Hongyang Chao. Building dual-domain representations for compression artifacts reduction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016.

[26] Bruce Hendrickson. Graph partitioning and parallel solvers: Has the emperor no clothes? In *International Symposium on Solving Irregularly Structured Problems in Parallel*, pages 218–225. Springer, 1998.

[27] Intricately. A look at netflix's cloud spend. Available at https://www.intricately.com/articles/netflix-aws-spend (2021/01/14).

[28] Kazuyoshi Itoh. Analysis of the phase unwrapping algorithm. *Applied optics*, 21(14), 1982.

[29] Cheolkon Jung, Licheng Jiao, Hongtao Qi, and Tian Sun. Image deblocking via sparse representation. *Signal Processing: Image Communication*, 27(6):663–677, 2012.

[30] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[31] Michael Kerrisk. The linux man-pages project, 2019.

[32] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pages 82–96. Springer, 2002.

[33] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimizedvia graph cuts? *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):147–159, 2004.

[34] Zhi Li, Christos Bampis, Julie Novak, Anne Aaron, Kyle Swanson, Anush Moorthy, and JD Cock. Vmaf: The journey continues. *Netflix Technology Blog*, 2018.

[35] AW-C Liew and Hong Yan. Blocking artifacts suppression in block-coded images using overcomplete wavelet representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):450–461, 2004.

[36] Jose L Marroquin and Mariano Rivera. Quadratic regularization functionals for phase unwrapping. *JOSA A*, 12(11):2393–2400, 1995.

[37] Mark W Miles. Interferometric modulation of radiation, January 6 2004. US Patent 6,674,562.

[38] GL Nayler, DN Firmin, DB Longmore, et al. Blood flow imaging by cine magnetic resonance. *J Comput Assist Tomogr*, 10(5):715–722, 1986.

[39] Yusuf Sani, Andreas Mauthe, and Christopher Edwards. Adaptive bitrate selection: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2985–3014, 2017.

[40] Jim Schwiegerling. Optical specification, fabrication, and testing. SPIE, 2014.

[41] Xiaojin Shi, Yunhua Zhang, and Jingshan Jiang. Insar image registration using modified correlation coefficient algorithm. In *2006 7th International Symposium on Antennas, Propagation & EM Theory*, pages 1–4. Ieee, 2006.

[42] NK Soni. Phase unwrapping algorithm using edge detection and statistical cost functions. 2012.

[43] Kim-Han Thung and Paramesran Raveendran. A survey of image quality measures. In *2009 international conference for technical postgraduates (TECHPOS)*, pages 1–4. IEEE, 2009.

[44] Ci Wang, Jun Zhou, and Shu Liu. Adaptive non-local means filter for image deblocking. *Signal Processing: Image Communication*, 28(5):522–530, 2013.

[45] Ren Yang, Mai Xu, and Zulin Wang. Decoder-side hevc quality enhancement with scalable convolutional neural network. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 817–822. IEEE, 2017.

[46] Jung Hum Yu, Linlin Ge, and Chris Rizos. Digital elevation model generation from interferometric synthetic aperture radar using multi-scale method. 2009.