Optimal Quality-Aware Predictor-Based Adaptation of Multimedia Messages

Steven Pigeon
pigeon@iro.umontreal.ca
Département d'informatique et de recherche opérationnelle
Université de Montréal

Stéphane Coulombe
stephane.coulombe@etsmtl.ca
Department of Software and IT Engineering
École de technologie supérieure
Université du Québec

Abstract

The Multimedia Messaging Service (MMS) platform allows messages composed of various multimedia attachments to be exchanged between users of mobile devices. However, these heterogeneous devices exhibit different capabilities regarding what media types, resolution, and maximal message size they support making the adaptation of messages mandatory to ensure compatibility between sending and receiving devices. The challenge is therefore to adapt messages so that they satisfy the receiving device's constraints in a way that both maximizes the user experience and minimizes the computational cost of adaptation. Minimizing computational cost will help cope with high-volume traffic while maximizing user-experience, as estimated by the perceived quality of adapted messages, will secure the service provider's user base. In this work, we propose a generic adaptation scheme based on predictors for file size and image quality resulting from transcoding parameters applied to a given image that will explicitly maximize perceived quality as estimated by the structural similarity image quality index. We will further show that our proposed method is resilient to the imprecision of the predictors and that it yields significantly better quality at a greatly reduced computational cost compared to other methods proposed in prior art.

1 Introduction

From the users' point of view, Multimedia Messaging Services, or MMS, merely provide a convenient mean of exchanging messages composed of various multimedia attachments such as audio, still image, and video, between mobile terminal users. If, for a user, sending a message is a simple matter, as it suffices him to assemble attachments and select recipients, the service provider's task is significantly more complex as he must ensure not only fast delivery of messages but also interoperability between his users' heterogeneous mobile devices [19].

Ensuring interoperability means that each attachment (and the message as a whole) is potentially adapted so that it satisfies all of the receiving terminal's constraints. Such constraints specify the maximum message size the device can receive and decode and other limitations such as the maximal image resolution and the types of media the device can interpret correctly. For image attachments (for the most part JPEG images taken from the mobile device's camera), adaptation will consist in adjusting resolution, compression parameters, and sometimes even file format, so that the resulting image is compatible with the receiving terminal's capabilities. Unfortunately, adaptation cannot be devolved to the receiving terminal because the standard precludes terminals from receiving larger messages than what they can handle as specified by their limitations [8, 9, 35]. Having the sending terminal adapt the message is also quite impractical as it supposes it has a description of the receiving terminal, which in practice it does not have, and in the case of multiple recipients, adaptation would be too intensive for a such device where battery life and computing power are limited. Therefore, adaptation *must* be performed server-side.

The task of adapting a single message with a few image attachments does not seem daunting given the relatively small message sizes allowed by the MMS standard (which we discuss in more details in the next section), especially when one considers a server-class computer to perform the adaptation. Even adapting a JPEG image to fit given constraints such as resolution and file size seems trivial. However, adapting an image to specific constraints is an intensive process, as estimating efficiently and accurately the file size resulting from a given transformation remains a challenge [20, 21, 39, 40]. Adapting a whole message to fit the receiving device's constraints so that it maximizes overall perceived quality remains a complex operation [41] that we will discuss at length in this chapter. Finally, one must realize that a service provider does not have to adapt the occasional message, but to adapt potentially very large numbers of messages in a timely fashion. One therefore concludes that adaptation must be performed rapidly in order to cope with high-volume MMS traffic, and that it must be done in a way that yields a superior user-experience, in particular yielding images with a high perceived quality.

Of course, one may be tempted to resort to trivial adaptation strategies, for example indiscriminately shrinking images to thumbnails and discarding troublesome media such as video, but this is clearly unacceptable as users expect not only the messages to be delivered in a timely fashion, but also to be adapted in a way that does not incur objectionable degradation. Other simple strategies, such as forwarding the messages to an e-mail account [10, 30] or sending an URL by SMS where the users can fetch the message via the device's browser [18], will also lessen the user experience by failing to provide the integration of services one expects from MMS capable devices. Therefore, the provider must perform the best possible adaptation of messages in order to maximize the user-experience of his customers, preferably doing so at the lowest possible computational cost.

Minimizing computational cost is a necessity beyond merely ensuring diligent delivery of messages between users. In 2009, MMS traffic grew 48% in the United States alone, where 34.5 billion multimedia messages were exchanged [27]. With a shift towards green technologies and energy-efficient data centers, it seems unrealistic to hope coping with such a traffic with an annual growth of nearly 50% using naïve adaptation methods and merely adding new servers (with all the complications it implies) to meet demand. The solution to mitigate this problem is to use better algorithms that perform message adaptation in a computationally efficient way without sacrificing user-experience.

It is in this context that we propose a novel and computationally efficient method of multimedia message adaptation that explicitly maximizes user-experience under the constraints of the receiving terminal. In this chapter, we will expose our solution where we propose to not only to maximize explicitly resulting quality of adapted messages using an efficient algorithm, but to speed-up optimization significantly by the use of *predictors*. Predictors are fast algorithms that predict the resulting file size and perceived quality of an image on which were applied *transcoding parameters*, parameters that describes how to transform the image—for example by specifying a scaling factor to change resolution and other parameters to affect the level of compression.

This chapter is structured as follows. In the next section, section 2, we introduce the reader to the environment of multimedia messaging services and its constraints. In section 3, we discuss the various approaches proposed in literature to address the problem of MMS adaptation as well as adaptation of media in other contexts. We also discuss the nature of quality measure and their essential role in adaptation. We expose the details of our proposed solution in section 4. Section 5 describes our simulations and presents the results obtained. We discuss and interpret the results in section 6. Finally, the chapter closes with section 7, where we summarize our contributions and present perspectives for future work.

2 The MMS Environment

In this section, we will cursorily introduce MMS messages and the MMS operating environment. We will discuss the challenges it poses for multimedia messages adaptation.

An MMS message is essentially a MIME e-mail message [13, 17, 23, 42] as defined by the MMS encapsulation specification [34] where one can attach a number of media files (with acceptable formats defined in [8] and [9]). The presentation of the media elements can be simple attachments, but they can be structured using HTML or XHTML [6, 15], or dynamically arranged in a slide-show-type presentation using Synchronized Multimedia Integration Language, or SMIL [4, 14, 28]. In other words, the MMS standard provides a number of means to ensure not only delivery of media attachment, but also that they are presented adequately.

Not only MMS-capable devices are heterogeneous (devices can be any one of the tens of thousands of different models of terminals or even a desktop computer), the MMS network itself is composed of a great number of different technologies. The mobile devices may use any of the many wireless technologies (GSM [2], GPRS, TDMA, CDMA, or 3G [1]) but fortunately, network specificities are abstracted through the Wireless Application Protocol (WAP) [33] that provides transport and through the Wireless Session Protocol that provides session control (for an introduction to the transaction-based protocol ensuring the delivery of the messages from the mobile device to the service provider's network servers, the reader is directed to [19]).

When a MMS message is sent, it is first processed by the service provider's server that determines how to route the message to its recipient. The server first determines if the recipient is local or if it belongs to another service provider. In the latter case, the message is forwarded to the recipient's service provider using a MM4 network, the inter-working between MMS service centers. If the recipient is local, the service provider finds himself with basically two cases to consider. The first, and usual, case is when a MMS is sent as a MMS to a recipient on a mobile device. The second case to consider is when an e-mail is sent to a MMS-only address. The third and fourth cases, (sending an MMS to an e-mail address and sending MMS-like e-mails), are not very interesting as, unlike the two first cases, they do not need significant adaptation: as a MMS is essentially a MIME e-mail, it can be send as-is and left for the e-mail client to display. The two first cases, however, may require adaptation as sending and receiving device may not have the same capabilities.

Device capabilities are standardized via *capability profiles*, or profiles for short. Table 1 present a simplified view of profiles, where constraints such as maximum image resolution, maximum message size, and supported image media format are listed. The profiles also specify the type of character encodings, presentations, and encapsulation each profile is to support, as well audio, speech, and video formats for attachments [30], but we will ignore these aspects in this work.

Therefore, when a MMS server receives a MMS message, it first *characterizes* it, that is, assesses its contents by examining presentation and individual attachments, and then determines if adaptation is needed given the receiving terminal's capabilities. To do so, the service provider may maintain a database associating recipients addresses to their corresponding de-

Profile Name	Maximum Resolution	Maximum Message Size	Image Media
Image Basic	160×120	30 KB	Baseline JPEG, GIF87a/89a, WBMP
Image Rich	640×480	100 KB	Baseline JPEG, GIF87a/89a, WBMP
Megapixel	1600×1200	300 KB	Baseline JPEG, GIF87a/89a, WBMP
Content Rich	1600×1200	300 KB	Baseline JPEG, GIF87a/89a, WBMP, SVG

Table 1: Simplified MMS profiles.

vices. The device characterizations are maintained in a database that describes the devices, beyond MMS capabilities, containing information about screen resolution and other features, information that could be used, in principle, for finer adaptation of the messages. One such database, the Wireless Universal Resource File (WURFL) maintained by Luca Passani, contains approximately 11300 devices [37].

Using the database, the provider is capable of devising the best strategy for message adaptation, whether adaptation or mere pass-through. In some cases, it may be possible to simply relay the message as-is because its contents are already compatible (as determined by characterization) with the receiving terminal (for example, a message using Image Basic sent to a Megapixel-capable device). In any other case, adaptation must be performed.

3 MMS Adaptation

The goal of MMS message adaptation is to make a message compatible to the recipient's receiving device. To do so, we may need to *transcode* images, that is to change their resolutions, compression parameters or even compression format altogether. However, changing compression format may not be wanted in general. First because most of the images are "natural images" in JPEG format (as taken from camera phones and JPEG is generally efficient in this case), and second because the remaining images types will likely need to remain in their original formats to retain their specificities—animations in GIF89a format, for example, will need to be preserved. For this reason, and ultimately without loss of generality as we will discuss in section 6, we will consider here only the case of all-JPEG message adaptation.

In subsection 3.1, we discuss prior art and other techniques relevant for the problem of MMS adaptation. In § 3.2, we discuss the problem of visual quality assessment and motivate our choice of quality metric, the structural similarity of Wang et al. In § 3.3, we establish the notation we will be using in the remaining of this work as well as present a formalization of the problem we address.

3.1 Prior Art

A priori, one would think that adapting a Baseline JPEG image [7, 38] against a given maximal file size is an easy task. One could think it suffices to adjust compression parameters until the desired file size is obtained, but for JPEG, especially at very high compression, that means introducing a host of conspicuous artifacts, such as color bleeding and blocking. Blocking occur when the compression is too aggressive and that DCT patches boundaries cease to match their neighbors' boundaries, and the discontinuities are visually displeasing [29, 52]. Color bleeding is a side effect of quantization in the usually sub-sampled chroma planes [56]. To avoid these artifacts and yet meet the target file size, one must rather consider a mixed strategy where resolution and compression are adjusted jointly, that is, also consider solutions where the resolution of the image is lowered but the image is compressed less aggressively, thus avoiding compression-related artifacts while reducing the overall file size significantly. Therefore, the best trade-off between resolution change and compression settings is to be found (while meeting the constraint of the desired file size), where "best" is measured by an objective quality measure highly correlated with the quality perceived by a human observer—we discuss the problem of the quality measure in § 3.2.

Some solutions were proposed to recompress and/or adjust resolution in the (partially) compressed domain, where operations are performed on the DCT coefficients resulting from a partial decompression. In Ridge's method [44], the file size resulting from a new compression parameter is estimated by computing the code-length of the DCT coefficients after requantization. The compression parameter is adjusted (using a binary search) until the parameter yielding the file size closest to, but not exceeding, the target file size is found. This method is more efficient than completely decompressing and recompressing the image six or seven times (as the main parameter controlling compression, known as the *quality factor*, can vary from 1 to 100, according to the Independent JPEG Group semantics [5], and $\log_2 100 \approx 7$) but it is still very intensive; it is unclear what are the speed-ups one can expect from such a method. Furthermore, Ridge's method of bitrate adaptation does not consider the case where resolution must be jointly optimized—although he does propose a (partially) compressed-domain resizing method.

Fast algorithms for resolution changes in the (partially) compressed domain were proposed [22, 32, 44]. These methods manipulate the DCT coefficients in order to merge four adjacent 8×8 tiles into a single 8×8 tile, effectively reducing the resolution by a factor of two in each direction. The operations necessary for the proposed power-of-two reduction algorithms are rather complex, and it is unclear how they compare, speed-wise, with a well-implemented classical pixel-domain filter. Additionally, these methods do not generalize well to arbitrary resolution changes, that is, scaling an image by a factor of, say, 0.9 (thus yielding an image with a resolution 90% of the original in both directions). To summarize, these algorithms are complex from a conceptual point of view, have unclear speed-up benefits (as speed-ups are reported only relative to the DCT operations, neglecting the remaining operations such as entropy (de)coding, which are not a negligible part of the operation), and are limited to reductions by powers of two.

Simultaneously adjusting scaling and quality factor in order to maximize perceived quality under the constraint of a maximal file size is therefore an expensive process if one proceeds by successive transcodings, even when using an efficient parameter-search method. As we will discuss in section 4, the parameter search is further complicated by the fact that the file sizes of each attachment in a message are added together for the total message size; and therefore parameters must be optimized jointly across all images, leading to a potentially very large number of combination examined. In this case, it is impractical to proceed to optimization by performing a very great number of actual transcodings. To speed-up optimization, we propose to avoid tentative transcodings by the use of predictors [20,21,39,40] for the file size and quality resulting from applying a scaling factor and a quality factor to an image. We discuss predictors in § 4.2.

*

* *

Image and media adaptation was, of course, considered before in other contexts. For example, the problem was considered for mobile browsing where clients with varying reception bandwidths access media on the web, and where media is adapted not to satisfy the devices' display constraints, but to maximize quality of service as defined by download time [24, 49]. In these settings, resolution and image quality are sacrificed in order to provide a fast, responsive browsing experience. In a sense, these solutions address only a part of the problem we consider, that is, the part where we determine the largest possible file size to meet a constraint of download time given a device's bandwidth, but without consideration for the visual quality of the images.

Adapting media with explicit maximization of quality was proposed in Mohan et al. [31,48]. They propose to maximize "content value" (a measure inversely proportional to distortion) rather directly visual quality, and the problem is formulated as a rate-distortion optimization problem. After observing that this particular optimization problem is difficult to solve directly if all possible transcoding parameters are considered, they adapt Shoham's et al. [47] quantization method to their framework, basically using a quantization of the parameter combination space they examine; that is, they will consider only a rather small number of precomputed profiles for optimization. In the paper, they present six profiles (from full resolution 24 bits image to "alt-text" description of the image) each corresponding to a generic device class such as a desktop computer or a PDA.

Others proposed techniques based on message understanding or region of interest extraction. In [55], Yan and Kankanhalli propose a model of MMS adaptation based on "multimedia simplification," where images are cropped around their most likely region of interest, and video clipped around their most salient moment as estimated using hints such as the loudness of its sound track. Chen et al. [16] propose an approach based on probable element relevance to simplify and adapt content of Web pages, discarding elements that are not immediately relevant to the requested topic. However, these techniques are deemed too computationally expensive to face the demands of high-volume transcoding, furthermore we may question their applicability for MMS. Indeed, it is unlikely that users would want their images cropped or video truncated, much less so if the system fails to provide optimal decisions every single time—from the *users*' point of view. Also, message simplification techniques based on relevance are probably inapplicable to an image-only MMS message containing possibly unrelated images chosen by the user; in a sense, for the user sending the message, the images are *all* related.

3.2 Image Quality Assessment

Accurately measuring perceived image quality—a crucial operation in media adaptation—remains problematic. The peak signal-to-noise ratio (PSNR) has been used extensively in literature, whether for sound or for image processing, but the objective measure of PSNR is not a good estimator of *perceived* quality of a transformed signal, and will leave us in the need of a better measure.

The PSNR is defined from the mean squared error (MSE) between an original signal $X = \{x_i\}_{i=1}^n$ and a reconstructed or transformed signal $\hat{X} = \{\hat{x}_i\}_{i=1}^n$. Between an original image X and a transformed image $\hat{X}$ (of width w and height h) the MSE is given by

$$\mathrm{MSE}(X, \hat{X}) = \frac{1}{wh} \sum_{i=1}^{w} \sum_{j=1}^{h} (x_{ij} - \hat{x}_{ij})^2 \tag{1}$$

and the PSNR by

$$\mathrm{PSNR}(X, \hat{X}) = 10 \log_{10} \frac{M^2}{\mathrm{MSE}(X, \hat{X})} = 20 \log_{10} \frac{M}{\sqrt{\mathrm{MSE}(X, \hat{X})}}\ , \tag{2}$$

where M is defined as the maximal value one x_j can take, for example 255 if $0 \leqslant x_j < 256$. While other variants define M as the maximum value found in either X or $\hat{X}$, using a fixed maximum allows to compare against a class of signals rather against a single pair. However, from the very formulation of eq. (1), it should be clear that the PSNR is not invariant to simple transformations that may not affect perceived image quality such as translations, biases (adding or subtracting a small constant from X), and in the specific case of images, modifying the colors by either reducing or enhancing color saturation. In fact, while PSNR will measure an image degradation, enhancing saturation and boosting contrast may result in an image *more* pleasing than the original image.

If PSNR is a poor indicator of image quality because it basically ignores the human psychovisual model, which we aught to use explicitly as it is user-experience one wants to maximize in our context, we must therefore look at other image quality models. There are many models that try to model the human psychovisual response to images (see [46] for a survey and analysis), but one has to find the adequate trade-off between accuracy (for example, as measured by a ranking test with the mean opinion score [43]) and the computational cost (and other logistic considerations) of the measure. One such measure is the structural similarity (SSIM) of images [53, 54].

The structural similarity is based on the premise that images that are non-objectionable transformations of an original image lie nearby the original image on an image local luminance-contrast manifold. Taking into account the properties of the proposed structural similarity image space, the distance between an original luminance image X and a transformed luminance image $\hat{X}$ is given by

$$\mathrm{SSIM}(X, \hat{X}) = \frac{(2\mu_X \mu_{\hat{X}} + c_1)(2\sigma_{X\hat{X}} + c_2)}{(\mu_X^2 + \mu_{\hat{X}}^2 + c_1)(\sigma_X^2 + \sigma_{\hat{X}}^2 + c_2)}\,,$$

where μ_X and $\mu_{\hat{X}}$ are the means of X and $\hat{X}$, respectively, and where σ_X^2, $\sigma_{\hat{X}}^2$ and $\sigma_{X\hat{X}}$, are the variance of X, $\hat{X}$ and the covariance of X and $\hat{X}$, respectively. The constants c_1 and c_2 are added for numerical stability. Wang *et al.* further propose the use of a version of SSIM defined at a point, where averages and variances are computed using a normalized 2D gaussian weighting function on a $w \times w$ window (with w odd, and $w = 11$ in [53]). The SSIM at each of the image locations (i, j), denoted here $\mathrm{SSIM}_{ij}(X, \hat{X})$, are pooled to yield the MSSIM index:

$$\mathrm{MSSIM}(X, \hat{X}) = \frac{1}{wh} \sum_{i=1}^{w} \sum_{j=1}^{h} \mathrm{SSIM}_{ij}(X, \hat{X})\,. \tag{3}$$

In this work, we will use the MSSIM as the estimator of perceived image quality.

*

* *

The definition of MSSIM lets us measure the impact of changing compression parameters on image quality, but does not directly allow for the changes in resolution. The operation of scaling will affect the pixel-size of an image and therefore makes the direct application of measures like PSNR or SSIM impossible.

To deal with this situation, one has essentially three choices. Let X be the original image, and $\hat{X}$ the transformed image on which the scaling factor $0 < z \leqslant 1$ was applied (for example, a scaling factor of $z = 0.3$ will yield an image $\hat{X}$ with resolution of 30% of that of X). One can therefore compare X with $\hat{X}$ by scaling back $\hat{X}$ at X's original resolution; one can compare with X scaled down (but not compressed) at $\hat{X}$'s resolution; or one can scale both to an intermediary resolution, say the device's maximal image resolution. The first option assesses quality against the original, and in a sense measures the best possible reconstruction. The second option measures the best possible reconstruction at the size of $\hat{X}$, which favors thumbnailing. The last option measures the best possible reconstruction for a specific receiving device. In this work, we have opted for the first option. The last option implies that one has to train a large number of predictors (which we discuss in § 4.2) to accommodate the various device screen resolutions.

3.3 Formalizing the problem

In the previous sections, we discussed the general setting of the problem we consider in this work, from the structure of an MMS message to broad

MMS environment, passing by prior art and a discussion on image quality assessment. We have given the reader a high-level view of the problem we address, but in this section, we will formalize the notation, the problem elements, and the problem itself.

*
*   *

A message $M = \{m_1, m_2, \ldots, m_n\}$ is composed of n images m_i, each with resolution $R(m_i) = (w_i, h_i)$, file size $S(m_i)$, and original quality factor $QF(m_i)$. A receiving device D, for our needs, is characterized by the maximum message size $S(D)$ and a maximum resolution $R(D) = (w_D, h_D)$, both dictated by the device profile. For example, a device D capable of Image Rich messages (see Table 1) would report $S(D) = 300\,\mathrm{KB}$ and $R(D) = (640, 480)$.

The transcoding parameters series $T = \{t_1, t_2, \ldots, t_n\}$, with $t_i = (q_i, z_i)$, where q_i is the new quality factor and $0 < z_i \leqslant 1$ is the resolution scaling factor, describes the transformations to apply to a message M. The transcoding parameters t_i are applied to image m_i using function $\mathcal{T}(m_i, t_i)$, which yields a new image with resolution $z_i R(m_i) = (z_i w_i, z_i h_i)$ that is compressed with quality factor q_i with file size $S(\mathcal{T}(m_i, t_i))$.

Let $Q(m_i, \mathcal{T}(m_i, t_i))$ measure the quality of transcoded image $\mathcal{T}(m_i, t_i)$ relative to original image m_i. As we discussed in § 3.2, whenever the resolution of the transcoded image differs from the original (that is, whenever $z \neq 1$), the transcoded image is scaled back to the original resolution for comparison.

We are therefore interested in finding the optimal transcoding parameters series T^* that maximizes an objective function $\mathcal{Q}(M, T)$ (which we define and discuss in § 4.1), that is, to solve

$$T^* = \arg \max_{T \in \boldsymbol{T}(M,D)} \mathcal{Q}(M, T) \tag{4}$$

where $\boldsymbol{T}(M, D)$ is the set of all transcoding parameter series T that satisfies the constraints

$$\sum_{i=1}^{n} S(\mathcal{T}(m_i, t_i)) \leqslant S(D) \tag{5}$$

and

$$z_i \max(w_i, h_i) \leqslant W_D \,, \\ z_i \min(w_i, h_i) \leqslant H_D \,, \tag{6}$$

for $i = 1, 2, \ldots, n$.

Eq. (4) expresses a generic objective function which we want to be indicative to perceived quality. Eq. (5) states that the sum of the transcoded images file sizes must be smaller or equal to the maximum message size for device D, while eqs. (6) express orientation-independent resolution constraints where the image can fit the device maximum resolution D in either portrait or landscape orientation. Let us note that eq. (5) ignores the cost of the presentation of the message which should be normally included in the optimization. To reflect the cost of the presentation layer (and of the rest of the message itself including headers), one could rewrite eq. (5) by replacing $S(D)$ with $S(D) - P(M, D)$, where $P(M, D)$ is the cost of the presentation of message M on device D, but we do not address the problem of adapting the presentation in this work.

4 Proposed Solution

In this section, we present the details of our proposed solution. In § 4.1, we discuss the choice of the objective function for the problem of multipart messages. In § 4.2, we discuss the need for predictors to speed-up optimization and we describe the predictors used in this work, the JQSP predictor. We also introduce the concept of *oracular predictors* and discuss their expected properties in relation to our proposed solution. In § 4.3, we revisit constraints and objective function, proposing modification to accommodate predictors. We discuss the optimization algorithm used to maximize the objective function in § 4.4. Lastly, we present and discuss comparative adaptation algorithms in § 4.5.

4.1 Objective Function

In § 3.3, we have proposed to use an objective function $\mathcal{Q}(M, T)$ that measures the quality of the message M transcoded using the transcoding parameters series T, without defining it. In § 3.2 we discussed image quality metrics and we chose SSIM (or more exactly, MSSIM, as given by eq. (3)), as a quality metric. However, as SSIM is essentially a local correlation factor between original and distorted images, it yields values on $[-1, 1]$, but the useful range will be $[0, 1]$, where 0 already corresponds to a perfectly uncorrelated image (and -1 would correspond to an *anti-correlated* image). Constraining the quality measure $Q(m_i, \mathcal{T}(m_i, t_i))$ on $[0, 1]$ will allow us use the objective function

$$\mathcal{Q}(M, T) = \prod_{i=1}^{n} Q(m_i, \mathcal{T}(m_i, t_i)) \,. \tag{7}$$

The proposed objective function in eq. (7) presents only one of an infinite number of possible objective functions, but several aspects makes it especially suitable for the task considered. If maximizing eq. (7) is not the same as maximizing average quality of the transcoded images, the expected average of the $Q(m_i, \mathcal{T}(m_i, t_i))$ increases as eq. (7) increases, and the expected variance necessarily decreases [26,50,51]. While maximizing eq. (7) is not the same as maximizing the average quality of the transcoded image, it will still prevent choosing solutions where there is a significant difference between the best image and the worst, thus forcing balanced solutions.

4.2 On Predictors

However, as we mentioned in § 3.1, actually performing a transcoding with given transcoding parameters to examine resulting file size and image quality is an extremely expensive process, and therefore it is impractical to maximize eq. (7) by performing an exponentially large number of transcodings. Rather than computing $S(\mathcal{T}(m_i, t_i))$ and $Q(m_i, \mathcal{T}(m_i, t_i))$ exactly by performing a transcoding, we will use predictors, $\widehat{S}(m_i, t_i)$ and $\widehat{Q}(m_i, t_i)$, both formulating their prediction from the characterization of the image m_i (such as its original file size $S(m_i)$, quality factor $QF(m_i)$, and resolution $R(m_i)$) and the transcoding parameters t_i.

In previous works we have proposed such predictors [20,21,39,40] but in this work, we will use the predictors presented in [21], to which we will refer here as the JQSP, the JPEG Quality and Size Predictor. The salient point of the proposed predictors is that predictions are learned, rather than engineered, from a large image corpus on which was applied a large number of transcodings. The specific experimental conditions and the constitution of the corpus are detailed in section 5 and in [21].

However, if the file size and quality predictors from [21] are well behaved, only using these predictors would validate the predictors themselves more than our proposed solution for MMS adaptation, of which we want to show the efficiency and stability. To establish the upper-bound of obtainable quality for the proposed predictor-based algorithm, in addition to the JQSP, we will use *oracular* predictors that return the exact file size and quality resulting from applying transcoding parameters to a given image. Of course, the oracular predictors perform their pythian prognostication by actually transcoding the image and observing the exact resulting file size and quality.

The oracular predictors are especially well suited to characterize the graceful degradation—or lack thereof—of the proposed algorithm to predictor error. Since they are exact, oracular predictors can be used to simulate predictors with any error characteristics. In the experiment we performed, described further in section 5, we used, in addition to the JQSP and the exact oracular predictor, predictors with gaussian relative errors of 1%, 2%, 5%, and 10%, 95% of the time. Relative error is computed as $|\hat{x} - x|/x$, for exact value x and predicted value $\hat{x}$.

4.3 Objective Function and Constraints, Revisited

Using the predictors, we re-write the objective function eq. (7) for message M and transcoding parameters series T as

$$\widehat{\mathcal{Q}}(M,T) = \prod_{i=1}^{n} \widehat{Q}(m_i, t_i) \tag{8}$$

where $\widehat{Q}(m_i, t_i)$ formulates a prediction on the resulting quality of image m_i on which were applied the transcoding parameters t_i. Constraints will be modified to accommodate predictors. The size constraint, eq. (5), will be re-written as

$$\sum_{i=1}^{n} \widehat{S}(m_i, t_i) \leqslant S(D)\,, \tag{9}$$

where $\widehat{S}(m_i, t_i)$ formulates a prediction on the resulting file size of image m_i on which were applied the transcoding parameters t_i. However, the constraints of eqs. (6) are left unchanged, as there is no uncertainty in the resulting image resolutions.

Then the problem becomes to find the optimal predicted transcoding parameter series

$$\widehat{T}^* = \underset{\widehat{T} \in \widehat{\boldsymbol{T}}(M,D)}{\arg\max} \ \widehat{\mathcal{Q}}(M, \widehat{T}\,)\,, \tag{10}$$

where $\widehat{\boldsymbol{T}}(M, D)$ is the set of all possible transcoding that (probably) satisfy the constraints of eqs. (9) and (6). The formation of $\widehat{\boldsymbol{T}}(M, D)$ is discussed in § 4.4 and section 5.

4.4 Optimization Algorithm

To solve eq. (10) (or eq. (7)) efficiently, we will need an efficient algorithm, as it is impractical to test a combinatorial number of parameters to find the optimal transcoding parameter series. Fortunately, the optimization problem can be formulated as a *distribution of effort* problem [25], a classical problem in operations research with known efficient algorithms, where a limited number of resources must be distributed at a number of points in order to maximize a gain function under given constraints. Here the resources spent correspond to the file size of images, the total budget of which is determined by the maximum message size for the target device, and the gain function is the overall message quality as estimated by the objective function eq. (8), under the additional constraints of satisfying the receiving device resolution.

The particular form of eq. (7) (and eq. (8)) makes the problem amenable to efficient optimization algorithms [36], and in particular to dynamic programming or A^* search. In the A^* formulation, the problem of distribution of effort is usually modeled as a graph [25] where, after having reached an intermediate state s_i (where attachments up to attachment m_i were solved), the possible successor states of s_i, the $s_{i+1,j}$, if reachable (transiting from state s_i to state $s_{i+1,j}$ does not violate the constraints), are connected by edges corresponding to the possible transcoding operations of attachment m_{i+1}. Therefore, the edge between the state s_i and state $s_{i+1,j}$ is labeled by a transcoding $(q_{i+1,j}, z_{i+1,j})$. One solves for the best path in this graph, the path that maximizes the objective function while being admissible, that is, satisfying all constraints.

To minimize optimization time, one must therefore limit the number of successor states to be examined, that is, minimize the size of $\widehat{\boldsymbol{T}}(M, D)$. In particular, $\widehat{\boldsymbol{T}}(M, D)$ cannot contain an infinite number of elements. This implies that the set of all tuples (q_{ij}, z_{ij}) for attachment m_i considered must be quantized to a limited, ideally rather small, number of possibilities. However, it is unclear how one would prune combinations of q and z when considered jointly; and in our experiments, after the initial quantization discussed in section 5, we limited ourselves to pruning the set of possible transcoding parameters for each attachment m_i by excluding all transcoding (probably) exceeding the device constraints, that is, we eliminated all transcoded with a predicted file size exceeding the maximum message size and all scalings exceeding the resolution of the device—all z such that $zR(m_i) > R(D)$.

After optimization, the algorithm yields $\widehat{T}^*$, the (probably) best transcoding to apply to message M to satisfy the constraints of device D. However, it may be that the predicted transcoding parameter series produce a message that exceeds the constraints of the receiving device. If the message exceeds the constraints of the receiving device, a scaling factor $0 \ll \alpha < 1$ is applied to the maximum message size and optimization is performed again. We rewrite the size constraint of eq. (9) at the r-th retry as

$$\sum_{i}^{n} S(\mathcal{T}(m_i, t_i)) \leqslant \alpha^r \, S(D) \, . \tag{11}$$

The initial optimization, with $r = 0$, yields eq. (9). In our experiments, we set $\alpha = 0.9$.

4.5 Comparative Algorithms

To compare results from our proposed algorithm, we will use algorithms inspired from the previous literature (which is rather thin for this particular topic of MMS adaptation). The first algorithm, "successive profiles", inspired by the fixed adaptation strategy of Mohan et al [31], will apply successively more restrictive profiles to all images until the transcoded message satisfies the receiving device constraints. For this algorithm, a profile defines both the maximum resolution of images and the quality factor with

Resolution	Quality Factors
640×480	90, 80, 70, 60
320×240	90, 80, 70, 60, 50
160×120	90, 80, 70, 60, 50, 40

Table 2: Combination of resolution and quality factors forming the profiles used for algorithm "successive profiles."

which they will be compressed. For example, a profile could limit the resolution to 640×480 and impose a quality factor of 90. The next profile, more restrictive, could impose the same resolution but a quality factor of 80, and so on. The profile considered for this algorithm are shown in table 2. We will see, in section 5 that it is not helpful to have a great number of profiles.

The second comparative algorithm, "successive scaling", will only reduce the images' resolution while using a fixed, but otherwise reasonable, quality factor of 85, until it yields a message that satisfies the device constraints. The algorithm proceeds as follows. First, for each image m_i, the largest allowable scaling factor $0 < z_i \leqslant 1$ such that $z_i R(m_i) \leqslant R(D)$ is found: in order words, the image is initially scaled to the maximum resolution acceptable for the device. Adaptation proceeds by adjusting, at iteration $r = 1, 2, \ldots$, a global parameter β_r (initially $\beta_1 = 1$) that is applied to every image so that the scaling factor, at step r, for image m_i, is $\beta_r z_i$, yielding an image with resolution $\beta_r z_i R(m_i)$. Assuming the scaling factor controls quadratically the file size (as a scaling of z does not yield a file $O(z)$ times smaller, but $O(z^2)$ times smaller), a reasonable adjustment β_{r+1} (for $r > 1$) is given by

$$\beta_{r+1} = \alpha_2 \sqrt{\frac{S(D)}{S_r}}\,, \tag{12}$$

where α_2 is an dampening factor (set to 0.95 for our experiments) to ensure that the algorithm stops rapidly after only a few iterations, and S_r is the size of the message obtained at step r. The adaptation terminates when a message satisfying the device constraints is produced.

Both comparative algorithms heuristically maintain a balance between image scaling and quality factors to produce satisfactory message quality—one by using predefined profiles, possibly tried in a decreasing expected message size order, the other by adjusting only resolution but keeping a good quality using a fixed quality factor. One could think of other heuristic adaptation strategies, for example, fixing the images' resolution to at most $R(D)$ (leaving smaller images' resolution unchanged) and increase the compression aggressiveness by progressively lowering the quality factor until a message meeting the maximum message side is created. This would lead to messages with images with very conspicuous blocking artifacts (resulting from an aggressive JPEG compression), even more so as the number of attachments grows and that the limited message budget is split across the many images.

5 Simulations & Results

In this section, we detail our experimental setup. In § 5.1, we discuss the image corpora used for the training of the predictors as well as for the formation of the test MMS messages. In § 5.2, we describe the adaptation experiment itself, the various operating conditions including a description of the test machine, and we present the results thus obtained.

5.1 Image Corpora, Predictors, and Test Messages

The image corpus used to train the JQSP is formed of 70 000 JPEG images obtained from the web in 2008, using a crawler using high-profile web sites as origination points [39]. Mainly because of confidentiality, it is impossible to sample messages from a MMS provider's traffic, and we deemed that good surrogate for MMS traffic would be a web crawler sampling web sites with user-submitted content. Ideally, a service provider would replicate the experiments in [20, 21, 39, 40] with a much larger number of images sampled from their actual traffic, possibly retaining information about the distribution of the number of attachments, average size and quality factor of original images, etc., none of which, most unfortunately, we had access to at the time of writing.

Each of the 70 000 images was subjected to 100 transcoding (corresponding to all combinations of all quality factors $q \in \{10, 20, \ldots, 100\}$ and scalings $z \in \{0.1, 0.2, \ldots, 1.0\}$) yielding 7 000 000 examples. Scaling was performed using a Blackman filter for its desirable properties [12], and both resizing and compression were performed using the Magick++ library [3]. The training procedure for JQSP is described in [21]. The JQSP predictor is designed to predict the optimal transcoding parameters (q, z) given a target file size, and not directly predict file size from transcoding parameters, contrary to other predictors presented in [39, 40].

The target file sizes chosen to query the JQSP were selected so that they were spread 5% apart in relative size for a given attachment; thus greatly limiting the number of parameters to examine without jeopardizing quality of adaptation. The oracular predictors where constrained similarly to the training conditions of the JQSP predictor, that is, limited to quality factors $q \in \{10, 20, \ldots, 100\}$ and scalings $z \in \{0.1, 0.2, \ldots, 1.0\}$); infeasible transcodings (file sizes or resolution exceeding the device constraints) were pruned from the optimization.

The proposed algorithm used $\alpha = 0.9$ in eq. (11), that allows retries if optimization fails to produce a message that satisfies the device constraints. The comparative successive profile algorithm used the profiles listed in table 2. The successive scaling method, described in § 4.5, used $\alpha_2 = 0.95$. Both constants α and α_2 were set arbitrarily to reasonable, yet likely non-optimal, values.

Test examples were also obtained from a web crawler but at a later time, in the fall of 2010 [41]. The resulting corpus contains 370 000 JPEG image images. For each of the 220 MMS test messages, five images were drawn at random from this second corpus, yielding messages with an average size of 1.4MB and average image resolution of 1140×838.

5.2 Adaptation Experiment

For the experiments, the target profile chosen was Image Rich (images limited to 640×480 and message size to 100KB, see table 1), thus requiring an average 15 : 1 reduction ratio for the simulated messages. This difference in original message size and target was deliberate as it seemed that consider-ing cases where only a moderate adaptation of, say $2:$ or $3:1$ would not prove our point as strongly.

The predictors used for the experiments were the JQSP predictors, an (exact) oracular predictor, and four additional oracular predictors with 1%, 2%, 5%, and 10% relative error 95% of the times, respectively.

On the messages formed, we proceeded to their adaptation by our proposed algorithm using the various predictors and by the two comparative algorithms, and measured key indicators of performance, namely capacity, the propension of the algorithm to use all of the available message budget; objective function, to measure how the algorithm maximizes overall quality as measured by eq. (7) (as it was observed *after* transcoding and not predicted, therefore eq. (8) was not applicable); average image quality as a measure of balance in the solutions; and lastly the number of actual transcoding performed by the various methods and the wall-time needed for adaptation. The test machine was an Intel T9600 64bits CPU at 2.8 GHz running Ubuntu Linux 10.04 LTS, the current version at the time of writing [41]. Scaling was performed using a Blackman filter [12] and transcoding performed using Magic++ [3] in single-threaded mode.

The capacity, the portion of the maximum message size used by the transcoded message, is shown for the different predictors and algorithms in Fig. 1. The objective function score for the different algorithms are shown in Fig. 2 and the average SSIM for each attachment is shown in Fig. 3.

Fig. 4 shows the objective function scores resulting from individual messages, each curve sorted in ascending order separately. The curves do no allow to compare the relative performance of adaptation algorithms on a same message, but do render the general behavior of the different algorithms and predictors. Fig. 5 presents a similar graph for the average message SSIM.

Fig. 6 shows the times in seconds for the dynamic programming algorithm using the JQSP predictor, the successive scalings and successive profiles algorithms for our single threaded implementation on our test machine described previously. Oracular times are not shown, as not applicable to an actual implementation (and since a great number of actual transcoding are performed, we have times two orders of magnitude greater). Lastly, Table 3 shows the av-

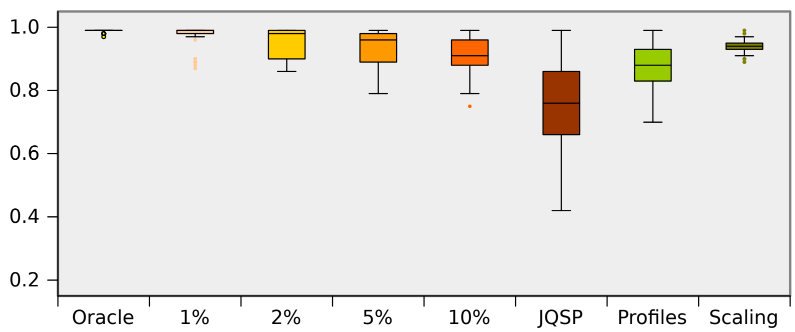

Figure 1: Message capacity distributions, by algorithms and predictors.

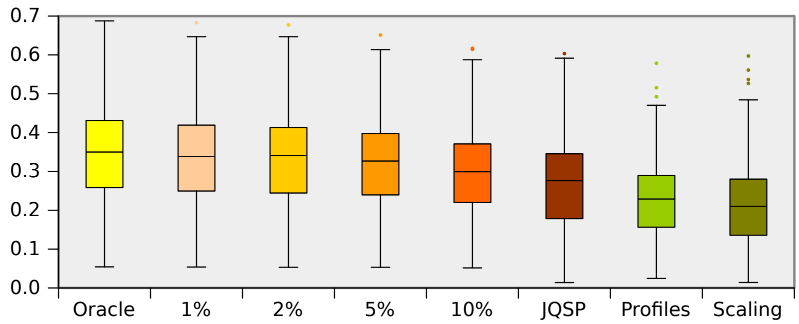

Figure 2: Objective function score distributions, by algorithms and predictors.

erage number of transcodings and retries (the number of times the algorithm must restart with new constraints) for different algorithms. Let us note that in our experiment, the fail rate is zero, as all transcodings are eventually successful. For a transcoding to fail, we would need to have an image large enough so that with a quality factor of 10 and a scaling of 10% it still exceeds the device constraints, and there were no such images in the test messages. If such a case would arise, one must think of a contingency method, possibly dropping attachments or splitting the message across many messages—issues we do not address in this work.

6 Discussion

One surprising thing, shown in figs. 1 and 2 is that is not sufficient to merely maximize capacity to achieve high adaptation quality; but that ca-

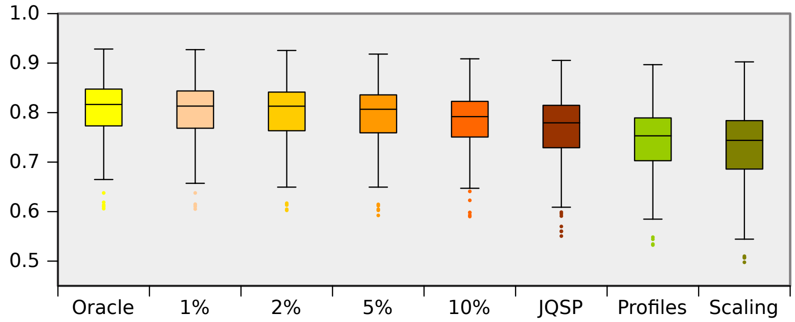

Figure 3: Average SSIM distributions, by algorithms and predictors.

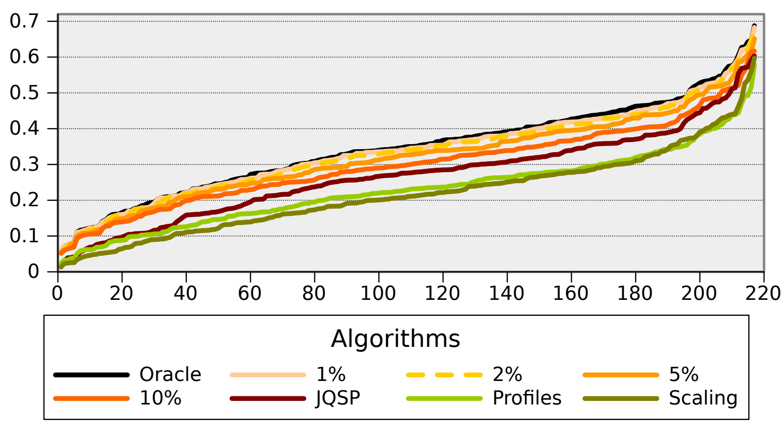

Figure 4: Objective function scores, by algorithms and predictors.

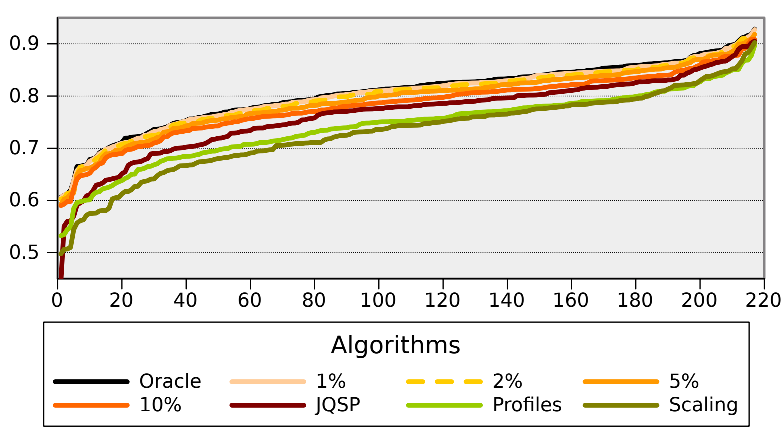

Figure 5: Average SSIM by message, by algorithms and predictors.

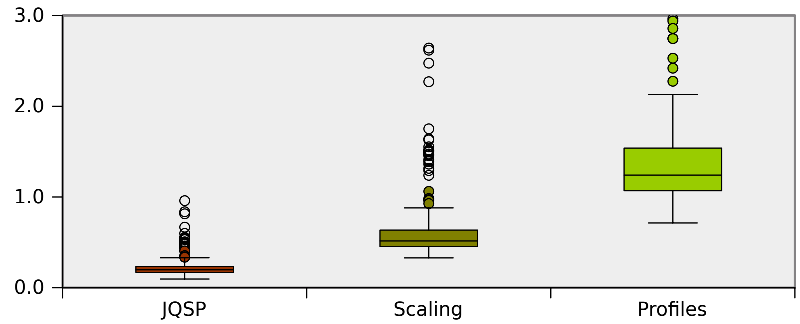

Figure 6: Message adaptation times distributions, by algorithm.

Optimization Algorithm	Transcodings	Retries	Objective Function
Oracle	5.00	0.00	0.35
1%	6.03	0.21	0.34
2%	6.54	0.31	0.33
5%	7.19	0.43	0.32
10%	8.25	0.65	0.30
JQSP	5.55	0.13	0.27
Profiles	33.36	5.67	0.22
Scalings	15.02	2.00	0.23

Table 3: Compared Averages of algorithms for a message of 5 attachments.

pacity must be maximized as a side-effect of quality-aware optimization. The great number of profiles used in the successive profiles algorithm and the efficient parameter search in the successive scaling algorithm allows them to get close to the capacity, but as neither explicitly maximize message quality, merely attempting to do so heuristically, the resulting quality is, in fact, inferior to our proposed method, all predictors considered.

Let us remark that maximizing eq. (7), the products of SSIM scores, for a message is not the same as maximizing the average SSIM score for the same message (although as discussed in § 4.1, both are strongly linked); but figures 4 and 5 confirm that the two are highly correlated, in particular, the ordering of algorithms and predictor pairs is preserved.

Figs. 2 and 5 show that the JQSP predictor behaves close to the 10% relative oracular predictor (probably closer to a 15% relative error predictor), and figs. 4 and 5 show that the predictor-based method using explicit quality maximization yields a quality significantly higher than the successive scalings and successive profiles algorithms. These figures illustrate that proposed algorithm is capable of graceful degradation with increasing predictor error, as quality and capacity decrease gracefully with increasing error rather than abruptly.

*

* *

The algorithmic complexity of the optimization methods is also interesting to study. The complexity of both the successive profiles and the successive scaling algorithms is dominated by the number of actual transcodings (see table 3), and their internal operations are negligible, almost null. Indeed, the decision part of the successive profiles algorithm reduces to a table look-up where profiles to apply are fetched and to determine whether or not an image is to be resized. For the successive profiles, the most complex part, excluding transcodings, is the computation of eq. (12) which only involves a handful of floating-point operations: nothing daunting for a server-class computer. For the successive profiles algorithm, the number of iterations is upper-bounded by the number of profiles in its table, while the successive scaling will converge quadratically fast, in a few iterations, being a Newton-Raphson-like method.

The cost of explicit optimization using dynamic programming is essentially $O(n\,|T(m,D)|^2)$ where n is the number of attachments and $|T(m,D)|$ is the expected number of transcodings for images m (therefore $T(m,D)$ denotes the set of all the possible transcodings of image m satisfying the device D). Solving using a blind depth-first search leads to complexity $O(|T(m,D)|^n)$, but using A^* with pruning (for example, cutting off search whenever a partial solution already exceeds constraints or does worse already than the best solution so far, which in turn implies that combinations are examined in an order that favors aggressive pruning), still leads to an algorithm with an exponential run-time worst case, but using admissible pruning, it can be made polynomial-time in $O(|T(m,D)|^\delta)$, where δ is a constant that does not depend on n but rather in the average depth explored [45, p. 85].

Fig. 6 shows that the run-time of explicit maximization is offset by the gain in fewer transcodings, as to be much faster, on average, with an average of 5.55 transcodings per message, than the successive scalings algorithm with an average of 15.02, and than the successive profiles algorithm with an average of 33.36, as shown in Table 3. The slight overshooting of the file size prediction for the JQSP predictor keeps the proposed algorithm from achieving maximal capacity, while still producing higher quality messages than the comparative algorithms, with the side-effect of keeping the number of retries lowest after the oracular predictor; a predictor that undershoots significantly would imply far more transcodings before reaching a solution that satisfies the device constraints.

Furthermore, the number of predicted transcoding parameters examined during optimization plays a non-negligible role in the proposed method performance. Even if the optimization algorithm is efficient, a great number of predicted transcoding parameters per image means a larger graph to explore and necessarily increased run-time, even with pruning, as run-time grows (at least) quadratically in the number of predicted transcoding parameters [36]. It then becomes a trade-off between the precision of the predicted transcoding and run-time. The successive scalings and successive profiles algorithms are also subject the speed/quality trade-off, but the results are far less interesting. The successive profiles algorithm could use fewer profiles, but already the resulting quality is inferior to the proposed algorithm. The same is true for the successive scalings method, which could start with an aggressive scaling factor of, say, $\beta_1 = 0.8$, but that would result only in even worse quality.

*

* *

The predictor-based solution we propose is not limited to JPEG-only messages. It can be generalized to any type of media by introducing new predictors for the desired media types. One could propose predictors for PNG, GIF, voice audio formats, etc., and the general framework of the solution would remain exactly the same. File size and resolution constraints are unchanged, except for the fact that the predictors now accommodate other media types; however, it may be necessary to add other constraints. Certainly one would need to consider the case where the compression format itself is to be changed.

The presentation itself, ignored in this work, can most likely be modeled as supplementary constraints or with minor changes to the optimization algorithm. For example, one could enforce proportionality (i.e., all pictures maintain relative size) by sharing the scaling factor across all images, as the successive scaling algorithm does. If the new constraints are not amenable to dynamic programming (as not all types of constraints or objective functions can be used [11]) one will need to use an A^* search which is, in many regards, a more general solution than dynamic programming.

7 Conclusion

In this work, we proposed a solution to MMS adaptation where the problem is explicitly modeled as an optimization problem where the goal is to maximize image quality (with MSSIM as a measure of user-experience) under the constraints dictated by the receiving mobile terminal. However, the novelty of our solution resides in the fact that we propose the use of predictors to speed-up optimization significantly without jeopardizing the resulting quality of adapted messages. We have shown, also, that the proposed method degrades gracefully with increased predictor error and that, overall, it performs much better, even with rather large predictor errors, than algorithms found in and inspired by the existing literature.

The proposed objective function, based on the structural similarity (MSSIM), is necessarily a simplification of the user-experience. Future work could explore how to model user-experience more appropriately and determine what objective functions would best replace eq. (7) as a measure of the goodness of the adapted message. One could also consider the use of better predictors, for example [40], which was not available at the time of writing [41]. Of course, the more accurately one can predict resulting file size and quality, the best the adaptation can be. Not only can we find better solutions, we can also use this knowledge to further prune optimization and obtain a faster adaptation algorithm.

Acknowledgments

This work was funded by Vantrix Corporation and by the Natural Sciences and Engineering Research Council of Canada under the Collaborative Research and Development Program (NSERC-CRD 326637-05).

References

[1] 3rd Generation Partnership Project. http://www.3gpp.org/.

[2] Global System for Mobile Communications. http://www.gsm.org/.

[3] ImageMagick and Magick++. http://www.imagemagick.org/.

[4] Synchronized Multimedia Integration Language (SMIL 3.0). http://www.w3.org/TR/SMIL3/.

[5] The Independent JPEG Group. http://www.ijg.org/.

[6] XHTML 1.1 – Module-based XHTML – Second Edition. http://www.w3.org/TR/xhtml11/.

[7] Recommendation T.81: Digital Compression and Coding of Continuous-Tone Still Images — Requirements and Guidelines. Technical report, CCITT, September 1992.

[8] 3GPP TS 23.140 V10.0.0. Multimedia Message Service (MMS) Media Formats and Codecs (Release 10), April 2011. http://www.3gpp.org/ftp/Specs.

[9] 3GPP2 X.S0016-000-A. 3GPP2 Multimedia Message System MMS Specification Overview, Rev. A, May 2003. http://www.3gpp2.org/.

[10] 3GPP2 X.S0016-330. 3GPP2 MMS MM3 Stage 3 for Internet Mail Exchange, June 2004. http://www.3gpp2.org/.

[11] Richard Bellman. *Dynamic Programming*. Dover, New York, 2003.

[12] R. B. Blackman and J.W. Tukey. *The Measurement of Power Spectra, from the Point of View of Communications Engineering*. Dover, 1959.

[13] N. Borenstein and N. Freed. RFC 1521: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanism for Specifying and Describing the Format of Internet Message Bodies. Technical report, IETF, September 1993.

[14] Dick C. A. Bulterman and Lloyd Rutledge. *SMIL 3.0 Interactive Multimedia for the Web, Mobile Devices and Daisy Talking Books*. Springer, 2008.

[15] Patrick Carey. *New Perspectives on HTML, XHTML, and XML*. Course Technology, 3rd edition, 2010.

[16] Rick. C. S. Chen, Stephen J. H. Yang, and Jia Zhang. Enhancing The Precision of Content Analysis in Content Adaptation using Entropy-Based Fuzzy Reasoning. *Expert Systems with Applications*, 37:5706–5719, 2010.

[17] David H. Cooker. RFC 822: Standard for the Format of ARPA Internet Text Messages. Technical report, IETF, August 1982.

[18] Nokia Corporation. How to Create MMS Services, V 4.0, June 2003.

[19] Stéphane Coulombe and Guido Grassel. Multimedia Adaptation for the Multimedia Messaging Service. *IEEE Communication Magazine*, 42(7):120–126, July 2004.

[20] Stéphane Coulombe and Steven Pigeon. Quality-Aware Selection of Quality Factor and Scaling Parameters in JPEG Image Transcoding. In *Procs. IEEE 2009 Computational Intelligence for Multimedia, Signal, and Video Processing (CIMSVP)*.

[21] Stéphane Coulombe and Steven Pigeon. Low-Complexity Transcoding of JPEG Images with Near-Optimal Quality Using a Predictive Quality Factor and Scaling Parameters. *IEEE Trans. Image Processing*, 19(3):712–721, March 2010.

[22] D. Dugad and A. Ahuja. A Fast Scheme for Image Size Change in the Compressed Domain. *IEEE Trans. Circuits and Systems for Video Technology*, 11(4):461–474, April 2001.

[23] R. Gellens. RFC 4356: Mapping Between the Multimedia Messaging Service (MMS) and Internet Mail. Technical report, IETF, January 2006.

[24] Richard Han, Pravin Bhagwat, Richard LaMaire, Todd Mummert, Véronique Perret, and John Rubas. Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing. *IEEE Personal Communications Magazine*, 5(6):8–17, 1998.

[25] Frederick S. Hilier and Gerald J. Lieberman. *Introduction to Operation Research*. McGraw-Hill Science, 9th edition, 2009.

[26] Tatsuo Ishihara. The Distribution of the Sum and the Product of Independent Uniform Random Variables Distributed at Different Intervals. *Trans. Japanese Soc. for Industrial and Applied Mathematics*, 12(3):197–207, 2002.

[27] J. Dwyer, III. MMS to Prosper as Mobile Marketing Becomes Mainstream. *Wireless Week*, January 2011.

[28] Tim Kennedy and Mary Slowinsky. *SMIL: Adding Multimedia to the Web*. SAMS Publishing, 2002.

[29] Ihor O. Kirenko and Ling Shao. Local Objective Metrics of Blocking Artifacts Visibility for Adaptive Repair of Compressed Video Signals. In *IEEE Int. Conf. on Multimedia and Expo*, pages 1303–1306, 2007.

[30] Gwenaël Le Bodic. *Mobile Messaging, Technologies and Services, SMS, EMS and MMS*. John Wiley & Sons, 2nd edition, 2005.

[31] Rakesh Mohan, John R. Smith, and Chung-Sheng Li. Adapting Multimedia Internet Content for Universal Access. *IEEE Trans. Multimedia*, 1(1):104–114, March 1999.

[32] Jayanta Mukherjee and Sanjit K. Mitra. Image Resizing in the Compressed Domain using Subband DCT. *IEEE Trans. Circuits and Systems for Video Technology*, 12(7):620–627, July 2002.

[33] Open Mobile Alliance. Wireless Application Protocol. http://www.openmobilealliance.org/Technical/wapindex.aspx.

[34] Open Mobile Alliance. Multimedia Messaging Services Encapsulation Protocol. March 2005. Open Mobile Alliance OMA-MMS-ENC-V1_2-20050301-A.

[35] Open Mobile Alliance. Enabler Test Specification for (Conformance) for MMS Candidate Version 1.3. Technical report, October 2010. Open Mobile Alliance OMA-ETS-MMS_CON-V1_3-20101015-C.

[36] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.

[37] Luca Passani. WURFL — The Wireless Universal Resource File. http://wurfl.sourceforge.net/.

[38] William B. Pennebaker and Joan L. Mitchell. *JPEG: Still Image Data Compression Standard*. Digital Multimedia Standards. Springer, 1992.

[39] Steven Pigeon and Stéphane Coulombe. Computationally Efficient Algorithms for Predicting the File Size of JPEG Images Subject to Changes of Quality Factor and Scaling. In *Procs. 24th Queen's University Biennial Symposium on Communications*, 2008.

[40] Steven Pigeon and Stéphane Coulombe. Efficient Clustering-based Algorithm for Predicting File Size and Structural Similarity of Transcoded JPEG Images. In *Procs. IEEE International Symposium on Multimedia (ISM)*, 2011. (to appear).

[41] Steven Pigeon and Stéphane Coulombe. Optimal Quality-Aware Predictor-Based Adaptation of Multimedia Messages. In *Procs. of Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, volume 1, pages 496–499, 2011.

[42] R. Resnik. RFC5322: Internet Message Format. Technical report, IETF, October 2008.

[43] Soroosh Rezazadeh and Stéphane Coulombe. Novel Discrete Wavelet Transform Framework for Full Reference Image Quality Assessment. *J. Signal, Image, and Video Processing*, pages 1–15, September 2011.

[44] J. Ridge. Efficient Transform-Domain Size and Resolution Reduction of Images. *Signal Processing: Image Communication*, 18(8):621–639, Sept. 2003.

[45] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 3rd edition, 2009.

[46] Hamid Rahim Sheikh, Muhammad Farooq Sabir, and Alan Conrad Bovick. A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms. *IEEE_TIP*, 15(11):3440–3451, November 2006.

[47] Yair Shoham and Allan Gersho. Efficient Bit Allocation for an Arbitrary Set of Quantizers. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36(9), September 1988.

[48] John R. Smith, Rakesh Mohan, and Chung-Sheng Li. Content-Based Transcoding of Images in the Internet. *Int. Conference on Image Processing (ICIP)*, 1998.

[49] John R. Smith, Rakesh Mohan, and Chung-Sheng Li. Transcoding Internet Content for Heterogeneous Client Devices. In *Int. Symposium on Circuits and Systems*, volume 3, pages 599–602, 1998.

[50] M. D. Springer and W. E. Thompson. The Distribution of Products of Independent Random Variables. *SIAM Journal on Applied Mathematics*, 14(3):511–526, May 1966.

[51] M. D. Springer and W. E. Thompson. The Distribution of the Products of Beta, Gamma and Gaussian Random Variables. *SIAM Journal on Applied Mathematics*, 18(4):721–737, June 1970.

[52] Luc Trudeau, Stéphane Coulombe, and Steven Pigeon. Pixel Domain Referenceless Visual Degradation Detection and Error Concealment for Mobile Video. In *Int. Conference on Image Processing (ICIP)*, 2011.

[53] Zhou Wang, Alan Conrad Bovick, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Processing*, 13(4):600–612, April 2004.

[54] Zhou Wang, Alan Conrad Bovick, and Eero P. Simoncelli. *Structural Approaches to Image Quality Assessment*, chapter 8.3, pages 961–974. Elsevier Academic Press, 2nd edition, 2005.

[55] We-Qi Yan and Mohan S. Kankanhalli. Multimedia Simplification for Optimized MMS Synthesis. *ACM. Trans. Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1), February 2007.

[56] Michael Yuen and H. R. Wu. A Survey of Hybrid MC/DPCM/DCT Video Coding Distortions. *J. Signal Processing*, 70:247–278, 1998.