Quality-Aware Predictor-Based Optimal Adaptation of Still-Image Multipart Multimedia Messages

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

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

***Abstract*—Multimedia Messaging Services (MMS) allow users with heterogeneous terminals to exchange structured messages composed of text, images, sound, and video. The MMS market grows rapidly, posing the problem of MMS adaptation, necessary to ensure interoperability between terminals. Message adaptation poses technological challenges, especially when one considers the case of high-volume service providing. In this work, we propose novel predictor-based dynamic programming approaches to MMS adaptation, explicitly maximizing user experience, rather than relying on heuristics to deliver satisfactorily adapted messages. We show that the proposed solutions lead to noticeably superior image quality with faster transcoding times than comparative algorithms inspired by what is found in actual products and in the literature.**

***Index Terms*—MMS, image adaptation, JPEG, optimization, predictor, dynamic programming, SSIM.**

I. Introduction

Multimedia Messaging Services (MMS) allow users with heterogeneous terminals to exchange structured messages composed of text, audio, still image, and video [1]. MMS traffic will likely grow very rapidly in the next few years, providing both business opportunities and technological challenges [2]. As MMS usage grows and standards evolve, providers are pressed to manipulate ever greater numbers of messages with richer media, but this daunting task cannot be reduced to mere message-passing as server-side adaptation of messages is necessary to ensure interoperability amongst users [3]. The very high volume of messages to be manipulated by service providers will ask for the most efficient adaptation algorithms and strategies to cope with demand, growth, evolution of standards, and to mitigate scaling problems.

For MMS applications, a receiving terminal is characterized by its capabilities—or maybe more accurately by its *limitations*—as defined by *profiles*. A profile determines the terminal's set of constraints, such as the maximum multimedia message size in bytes, the media types that the terminal can interpret as well as the specific constraints of individual media types such as maximum image size or maximum bit-rate [1]. A device supporting the "Image Rich" profile will support various still image formats, images with resolutions up to 640×480 pixels, and a maximum message size of 100 KB.

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). E-mails: pigeon@iro.umontreal.ca, stephane.coulombe@etsmtl.ca

Server-side adaptation will ensure that not only each individual multimedia attachment is compatible with the receiving terminal but also that the message as a whole can be sent and correctly interpreted. Each attachment is to be characterized and transformed, if need be, to satisfy the receiving terminal's constraints, whether by adjusting its format, resolution, or bit-rate. In the absence of server-side adaptation, a message exceeding the terminal capabilities (either by message size or media type) can result in terminal-specific behavior that ranges from merely incomplete messages to terminal crash. If the server is capable of determining the capabilities of the terminal but incapable or unwilling to perform adaptation, it can revert to alternate strategies such as sending a text-only SMS (Short Message Service) with the location of the original message for the user to download or browse by other means [4]. While this ensures delivery of content, it does not provide the user with a satisfactory experience, hinting that server-side adaptation is preferable from the users' point of view.

Multimedia message adaptation, however, is not a trivial task as it does not suffice to apply heuristics such as successively reducing the bit-rate, or successively reducing the resolution of the various message attachments until a message satisfying the receiving terminal's various constraints is produced to yield satisfactory results. We have shown, in previous work, that strategies using joint adaptation of compression parameters and scaling produced significantly better results than using either alone [5]. In this work, we propose to extend these ideas to the explicit optimization of compression parameters and scaling over a complete image-only message in order to achieve an adaptation that not only satisfies the receiving device constraints but also maximizes overall perceived quality, and therefore, we expect, maximizes user experience. While the techniques proposed in this work can be applied to any type of images, and to any media in general, we will, however, without loss of generality, restrict ourselves to the case of JPEG images, as most of the traffic is now composed of images from camera phones in JPEG format.

A priori, adapting a single JPEG image to fit given constraints appears as a trivial task, but JPEG adaptation is in fact a costly process, especially when one considers the high vol-

umes of images to manipulate in the context of server-side services. Of course, solutions have been proposed to speed up transcoding under constraints such as maximum file size and maximum resolution. Some solutions address the problem of estimating the transcoded file size of a single JPEG image but are still computationally intensive (in addition to necessitating extensive modifications in the existing JPEG software libraries) or overly rigid, for example, considering only scalings by power of two because they can be performed efficiently in the transform (DCT) domain [6]–[8]. Other solutions for image transcoding were proposed in the broader context of web or low-bandwidth resource access, and if some solutions are designed to transcode an image so that it fits the constraints while minimizing transcoding time [9], others use a small number of fixed transcoding profiles setting both compression parameters and maximum image resolution to achieve adaptation [10]; neither quite expressing the problem in terms of explicit maximization of resulting image quality or user experience. More complex adaptation approaches, based on the understanding of message contents and image points of interest were also proposed [11], [12], but, while promising, these techniques may be too computationally expensive for the type of high-volume transcoding needed by multimedia messaging service providers [2].

Therefore, adapting an image, even in JPEG format, against maximum file size and resolution while maximizing perceived quality in a computationally efficient manner remains a challenge as there are no established methods to estimate the resulting file size and quality of an image subject to changes in compression parameters and resolution. To this end, we have proposed, in previous work, predictors and systems to adapt images and messages [5], [13], [14]. In this work, we extend these systems and a previous work [15] to the adaptation of messages against receiving terminal constraints composed of multiple images using optimization algorithms maximizing explicitly perceived quality of the resulting messages. However, optimizing adaptation exactly is a very costly process and, to speed it up significantly, we propose to use predictors. In particular, we will reuse the predictor presented in [5], hereafter denoted JQSP (for JPEG Quality and Size Predictor). We will further show that not only the proposed methods yield better perceived quality than approaches found in, and inspired by, prior art, but also that the transcoding times are significantly lower.

The work is organized as follows. Section II lays out the basic definitions, details the problem space, and presents the proposed solutions. Section III describes the test methodology, the predictors, and algorithms used for comparison. The test results of the proposed algorithms under various conditions are presented in section IV, and discussed in section V. Finally, section VI concludes the present work.

II. Proposed Solution

Measuring perceived visual quality of images subject to transformations is a difficult task. The peak signal-to-noise ratio (PSNR) have been used in literature as an estimate of perceived quality but have been shown to be poorly correlated with perceived quality [16], in addition of being intolerant of transformations such as translation, scaling, and contrast—none of which automatically translates into a degradation of quality to a human observer. For the purpose of estimating the resulting quality, we will use the structural similarity index (SSIM) proposed by Wang *et al.* which exhibits tolerance to moderate transformations, while being better correlated to mean opinion scores (MOS) [17]. The SSIM is essentially a windowed correlation factor between original and distorted images and, as such, yields results on $[-1,1]$, but for our application we will constrain the measure on $[0,1]$, with any negative values mapped onto 0. Constraining the SSIM on $[0,1]$ allows us to use the objective function

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

where $M=\{p\,;\,m_1,m_2,\ldots,m_n\}$ is the message to be adapted, with presentation p and composed of the n images m_i, with resolutions $R(m_i)=(w_i,h_i)$, and where $T=\{t_1,t_2,\ldots,t_n\}$ is the series of transcoding parameters to be applied to the images. The transcoding parameters $t_i=(q_i,z_i)$ are such that $0\leqslant q_i\leqslant 100$ is the output quality factor (using the semantics proposed by the IJG [18]) and $0<z_i\leqslant 1$ a scaling factor used to resize the image. The function $\mathcal{T}(m_i,t_i)$, the transcoding function, applies transcoding parameters t_i on image m_i yielding an image with resolution $zR(m_i)=(z_iw_i,z_ih_i)$ compressed with quality factor q_i.

Lastly, $0\leqslant Q\left(m_i,\mathcal{T}(m_i,t_i)\right)\leqslant 1$ compares the original image m_i to its transcoded version $\mathcal{T}(m_i,t_i)$ using SSIM, as previously discussed. Since the images m_i and $\mathcal{T}(m_i,t_i)$ may differ in resolution, that is, whenever $z\neq 1$, the image $\mathcal{T}(m_i,t_i)$ is rescaled to the original resolution $R(m_i)$ before comparison [5]. In all cases, the image scaling is performed using a Blackman filter, chosen because of its spectral properties [19].

Eq. (1), as a measure of the quality of the transcoded message, is to be maximized by a transcoding operation series T under the constraints of a device D. The first constraint is that the total size of the transcoded images (plus the presentation meta-data) does not exceed the maximum size allowed for a message by the device. The second constraint is a resolution constraint, where all the images must have a resolution inferior or equal to the device's maximum resolution.

The size constraint is expressed by

$$\sum_{i=1}^{n} S\left(\mathcal{T}(m_i,t_i)\right)\leqslant S(D)-P(M,D)\,, \tag{2}$$

where $S\left(\mathcal{T}(m_i,t_i)\right)$ is the file size in bytes of the transcoded image $\mathcal{T}(m_i,t_i)$, $S(D)$ the maximum message size in bytes for device D, and $P(M,D)$ is the size, also in bytes, of the presentation (headers, markup, etc.) necessary for the adapted message M to display correctly on device D. The left-hand side of eq. (2) determine the *capacity*, the portion of the allowable byte budget used by the message using transcoding parameter series T. Let us also note that we do not interest ourselves in the quantity $P(M,D)$, the amount

of bytes necessary for the presentation of message M on device D, which would require adaptation of the presentation to estimate correctly. It is rather supposed to be a small, essentially negligible, part of the message budget.

The image resolution constraints for message M and device D are given by the orientation-independent resolution constraints

$$\begin{aligned} z_i \max(w_i, h_i) &\leqslant \max(w_D, h_D) \,, \\ z_i \min(w_i, h_i) &\leqslant \min(w_D, h_D) \,. \end{aligned} \tag{3}$$

where (w_D, h_D) is $R(D)$, the receiving device's maximum image resolution, which is independent of the device's actual screen resolution. We therefore suppose that the receiving device scales and rotates the pictures on screen for best viewing conditions.

The optimal series of transcoding T^* for a message M and a device D is therefore given by

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

where $\boldsymbol{T}$ is the set of all possible series of transcoding parameters satisfying the constraints of eq. (2) and eqs. (3). The cardinality of $\boldsymbol{T}(M, D)$ can be very large (even infinite) if we do not constrain the transcoding parameters to a rather small set of discrete values to avoid the combinatorial explosion of the number of states the algorithm solving eq. (4) examines. The quality factor, as defined by the IJG, is an integer taking values from 0 to 100, inclusive, but the scaling factor is a real, or at the very least rational, number and can therefore take an infinite number of values on $]0, 1]$. To solve this problem, in previous work [5], [13], [14], we have constrained both transcoding parameters to take at most ten distinct values, that is, quality factors were limited to the set $\{10, 20, \ldots, 100\}$, and scalings to $\{0.1, 0.2, \ldots, 1\}$, thus limiting the number of possible transcodings to 100. We will use the same values for the present work's experiments.

The objective function eq. (1) presents only one of the possible measures of overall message quality; however, several aspects make it especially suitable for the task considered. While maximizing eq. (1) is not the same as maximizing the average quality of the transcoded images, the expected average quality necessarily increases with increases of eq. (1), as a consequence of

$$\prod_{i=1}^{n} Q\left(m_i, \mathcal{T}(m_i, t_i)\right) \leqslant \min\left\{Q\left(m_i, \mathcal{T}(m_i, t_i)\right)\right\}_{i=1}^{n} \,,$$

where $\{x_i\}_{i=1}^{n}$ denotes the sequence $\{x_1, \ldots, x_n\}$. The nature of Q will furthermore cause the maximization of eq. (1) to reduce its variance [20]–[22]. Therefore, maximizing the proposed objective function will have the side effects of finding solutions with higher expected average quality and lower variance. Lower variance is especially interesting as it translate into reducing the risk of finding solutions were one transcoded image is of very poor quality while all the others are of good quality, rather than finding solutions where quality is balanced between all images.

The objective function eq. (1) is convex and separable, making it a Bellman equation [23], and thus amenable to efficient optimization using dynamic programming. More specifically, the optimization problem considered in eq. (4) is a *distribution of effort* problem, where a finite quantity of resources are allocated strategically in order to maximize a gain function [23], [24]. For the current problem, the resources are transcoded file sizes whose sum is constrained by the maximum message size as stated in eq. (2); the gain from allocation is the quality obtained as determined by the objective function, eq. (1); under the additional constraints of maximum resolution, as given by eqs. (3). This general class of problem was studied extensively and there exist efficient polynomial-time algorithms to find which allocation of resource maximizes the objective function under the given constraints [24], [25].

Solving eq. (4) exactly is possible, but would require for all transcoding parameters examined by the algorithm that an actual transcoding is performed to measure resulting file size and quality, clearly a prohibitive process. But rather than performing a transcoding for every combination of transcoding parameters examined, we will resort to fast predictors that, given a (superficial) characterization of an image m (such as original file size, quality factor, and resolution) and transcoding parameters t, will predict the resulting file size and quality of $\mathcal{T}(m, t)$, the transcoded image m to which were applied transcoding parameters t.

We have presented such predictors in previous works [5], [13] and in this study, we will use the file size and quality predictor presented in [5], which will be denoted JQSP, the JPEG Quality and Size Predictor. To assess the proposed methods' resilience to prediction error, we will, in addition to JQSP, use oracular predictors, predictors with known characteristics, discussed in the next section, section III.

The objective function using predictors is rewritten as

$$\widehat{\mathcal{Q}}(M, T) = \prod_{i=1}^{n} \widehat{Q}(m_i, t_i) \,, \tag{5}$$

where $\widehat{Q}(m_i, t_i)$ is the quality predictor taking an image m_i (or more exactly, its characterization composed of its original quality factor, file size, and resolution) and a transcoding operation t_i, rather than using $Q(m_i, \mathcal{T}(m_i, t_i))$ that compares the actual transcoded image $\mathcal{T}(m_i, t_i)$ with the original image m_i.

The size constraint must also be modified to accommodate predictors. The size constraint eq. (2) is rewritten as

$$\sum_{i=1}^{n} \widehat{S}(m_i, t_i) \leqslant S(D) - P(M, D) \,, \tag{6}$$

where $\widehat{S}(m_i, t_i)$ denotes the predictor of the size of image m_i on which were applied the transcoding parameters t_i. Eqs. (3), however, is unchanged as resolution remains a deterministic function of z_i and $R(m_i)$, and therefore contains no uncertainty.

The optimal predicted transcoding $\widehat{T}^*$ is given by

$$\widehat{T}^* = \underset{\widehat{T} \in \widehat{\boldsymbol{T}}(M,D)}{\arg\max}\, \widehat{\mathcal{Q}}(M, \widehat{T}) \,, \tag{7}$$

where the $\widehat{T}$ are series of transcoding parameters drawn from the set $\widehat{\boldsymbol{T}}(M,D)$ of all series of transcoding parameters on message M that (probably) satisfy constraints eqs. (3) and eq. (6) of the device D.

To summarize, we formulate the problem of adapting image-only MMS as a distribution of effort problem amenable to dynamic programming. We further propose to reduce the complexity of the problem by replacing actual transcodings by size and quality predictors in order to perform the optimization efficiently. The next section discusses the generation of the set $\hat{\boldsymbol{T}}(M,D)$, the series of transcodings on message M that (probably) satisfy the constraints of the device D. We will also present the details of the proposed optimization algorithms and the details of the two comparison algorithms.

III. Transcoding Algorithms

The JQSP (for JPEG Quality and Size Predictor) introduced in [5] differs from the predictor introduced in [13] as it does not directly predict file sizes and quality from an image characterization and a transcoding operation but rather predicts transcoding parameters and resulting quality from an image characterization and a target file size. Either predictors can be used to create a set $\widehat{\boldsymbol{T}}(M,D)$ for a message M and receiving device D, but in this work, as we mentioned earlier, we will use the JQSP. The JQSP predictor was trained on approximately 70 000 JPEG images gathered from the Internet, using a web crawler starting at various popular sites in 2008 [13]. The density of the JQSP predictions was adjusted so that target file sizes were set 5% apart, thus limiting the size of $\widehat{\boldsymbol{T}}(M,D)$ for optimization.

Proposing optimization methods based on a specific predictor such as the JQSP validate the predictor more than it validates the methods themselves. In order to show the properties of the proposed methods such as resilience to predictor error and ultimately determine the upper-bound on attainable quality, we will use *oracular* predictors. The oracular predictor "predicts" the exact file size and quality resulting from transcoding parameters applied to an image by actually performing the transcoding corresponding to the transcoding parameters. To further demonstrate that the proposed methods are resilient, that is, degrade gracefully in the presence of increasing errors in the predictors, we will use predictors derived from the oracular predictor with relative gaussian error on file size and quality of 1%, 2%, 5% and 10%, 95% of the time. To populate $\widehat{\boldsymbol{T}}(M,D)$ using the oracular predictors, the transcoding parameters were limited to quality factors of $\{10,20,\ldots,100\}$ and scalings of $\{0.1,0.2,\ldots,1.0\}$, thus limiting the number of transcoding parameters to at most one hundred per image. Let us now expose the details of our two proposed solutions to the problem of adaptation of JPEG-only messages, followed by the two comparison methods.

A. Dynamic Programming

The first proposed method, hereafter referred to as simply "dynamic programming", is to solve eq. (7) directly by dynamic programming to obtain the predicted optimal transcoding parameters series. The optimal transcoding parameters found by explicit optimization are then used to perform message adaptation. If the adaptation yields a message larger than the maximum message size for the receiving device due to size prediction error—it cannot yield a message with images of incompatible resolutions—a new, smaller, maximum message size is set for the device and adaptation is retried. The maximum message size is adjusted by a factor α_1 (set to 0.95 in our experiments) at each iteration, that is, we rewrite the size constraint eq. (6) as

$$\sum_{i=1}^{n}\widehat{S}(m_i,t_i)\leqslant\alpha_1^{k-1}S(D)-P(M,D) \tag{8}$$

for the kth iteration; at the first iteration, $k=1$, the constraint is equivalent to eq. (6).

B. Step Dynamic Programming

However, as we will discuss further in section V, the experiments show that using dynamic programming as described above, prediction error cumulate rather than cancel out—especially when the predictor is biased. The second proposed method, that we will refer to as "step dynamic programming" mitigates error propagation by proceeding by iterative refinement of the solution, again based on dynamic programming. Step dynamic programming will first optimize the message globally and determine the predicted optimal transcoding parameters series, but will transcode only the first image (in attachment order). After the first image is transcoded, its actual file size is observed and the budget for the remaining images is readjusted to take into account the transcoded image—and the corresponding prediction error. The remaining images are optimized jointly and, again, only the first of the remaining images is transcoded, its transcoded size observed, and budget readjusted. That is, eq. (8) becomes

$$\sum_{i=1}^{s-1}S(\mathcal{T}(m_i,t_i^*))+\sum_{i=s}^{n}\widehat{S}(m_i,t_i)\leqslant\alpha_1^{k-1}S(D)-P(M,D)$$

at step s (with $s=2,3,\ldots,n-1$, at $s=1$, we use eq. (8)) of the kth (with $k>1$) iteration. The t_i^* denote the transcoding parameter already chosen (but not necessarily optimal in an absolute sense). The next step of optimization proceeds by solving the modified objective function on the last $n-s$ terms, corresponding to the images yet to be transcoded. The objective function eq. (5), at step $s=2,3,\ldots,n-1$, becomes

$$\widehat{\mathcal{Q}}_s(M,T)=\left(\prod_{i=1}^{s-1}\mathcal{Q}\left(m_i,\mathcal{T}(m_i,t_i^*)\right)\right)\left(\prod_{i=s}^{n}\widehat{Q}(m_i,t_i)\right), \tag{9}$$

where the left part corresponds to images already transcoded (and therefore with known quality—which we do not necessarily need to observe) and the right part correspond to the objective function to maximize.

Readjusting budget and re-optimizing at each transcoded image greatly reduces the propagation of prediction errors, with the consequence that, as we will see in section IV, the algorithm makes better use of capacity.

TABLE I
COMBINATION OF RESOLUTION AND QUALITY FACTORS FORMING THE PROFILES USED FOR ALGORITHM "SUCCESSIVE PROFILES."

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

C. Comparative Algorithms

To compare the two proposed and novel solutions, we will use two algorithms inspired by the fixed profiles adaptation strategy of Mohan *et al.* [10]. The first comparative algorithm, "successive profiles," 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 quality factor and maximum resolution. For example, a profile could limit the resolution to 640×480 with a quality factor of 90. The next profile could use the same resolution but a quality factor of 80, the next could reduce the resolution to 320×240 but keep a quality factor of 80, and so on; each successive profile being more restrictive than the preceding, reducing resolution, quality factor, or both. The number and determination of useful profiles in this context depends on the performance objectives one expects. The profiles used for our experiments are shown in Table I. However, we will see in the next section that it is not useful to merly have a great number of profiles.

The second comparative algorithm, "successive scaling," will adjust only the images' scalings while using a fixed, but reasonable, quality factor of 85. Starting at the maximum resolution allowable for the receiving device, the algorithm will successively scale down all images (using the fixed quality factor) until the adapted message satisfies the device constraints. For each image m_i, the largest allowable scaling factor $0<z_i\leqslant1$ such that $z_iR(m_i)\leqslant R(D)$ is found. Adaptation proceeds by adjusting, at iteration $k=1,2,\ldots$ a global parameter β_k (initially $\beta_1=1$) that is applied to every image so that the scaling factor of image m_i at step k is $\beta_k z_i$, yielding an image of resolution $\beta_k z_i R(m_i)$. As scaling controls quadratically the file size (a scaling z yields an image of relative surface z^2), a reasonable adjustment β_{k+1} (for $k>1$) is given by

$$\beta_{k+1}=\alpha_2\sqrt{\frac{S(D)-P(M,D)}{S_k}}\,,$$

where α_2 is a dampening factor (set to 0.95 for our experimentation) to ensure that the budget is reduced further at each iteration, as well as limit the number of iterations, $S(D)$ is the maximum message size for the receiving device D, $P(M,D)$ is the size of the presentation of message M on device D, and S_k is the size of the message obtained at step k. The adaptation terminates when a message satisfying the device constraints is produced.

Both comparative algorithms try to heuristically maintain a reasonable balance between image scaling and quality factors—one through the predefined profiles and the other by using a fixed but relatively high quality factor while adjusting only resolution—in order to provide better image quality. One could also think of an algorithm where the images are reduced to $R(D)$, the maximum resolution for the receiving device and where further adaptation is achieved only by using successively coarser quality factors until the message satisfies the receiving device constraints. That strategy would, however, lead to the undesirable result of relatively high resolution images compressed with very low quality factors, thus exhibiting conspicuous blocking artifacts. This algorithm would likely produce worse results than either of the proposed comparative algorithms, and as such is not very interesting.

IV. RESULTS

For the experiments, we created four groups of 1000 MMS, with two (a minimum for "multipart") to five attached images. The images with resolutions between 320×200 and 3000×2000 were randomly chosen from a database of 370 000 images obtained by crawling the web in the fall of 2010 [15], rather than the database from previous works from [5], [13]. The profile chosen to test adaptation in our experiments is "Image Rich" (supporting images with resolutions up to 640×480 and a maximal message size of 100 KB). Forcing messages to "Image Rich" from the original MMS (whose average message size of 284 KB, 563 KB, 790 KB, 1.2 MB, and 1.4 MB, for 1, 2, 3, 4, and 5 attachments respectively) demonstrates that the different algorithms tested are put to stress with adaptation ratios up to $\approx14:1$.

The MMS in all groups were transcoded using the compared algorithms. All experiments used the same series of MMS, and the oracular predictors with gaussian noise (described in section III) used the same seed (and therefore the same pseudo-random sequence). Further, in all compared methods, we scaled images using a Blackman filter [19], with the actual image processing performed by ImageMagick's Magick++ library [26]. The experiments were performed on a Dell PowerEdge R210, with an Intel i3 540 CPU running at 3.07GHz, 4GB RAM, Ubuntu 11.04 with kernel 2.6, Magick++ 6.6.2, and G++ 4.5.2., a plausible setup for a transcoding node.

Tables II to V summarize the experimental results, showing, for each combination of number of attachments, optimization algorithms, and predictor the resulting number of transcodings performed, the average number of retries, the capacity, the average quality of the transcoded images, and the objective function score. Figures 1 and 2 show the distribution of resulting capacity and image quality, respectively, for 5 attachments using box-plots. Finally, table VI presents the average transcoding times (in seconds) for both proposed algorithms using the JQSP predictor versus the two comparative algorithms. Figure 3 shows the distribution of transcoding times for 5 attachments, also using the JQSP predictor (oracular times are excluded from the results as oracular predictors perform a great number of transcodings in order to formulate their "predictions").

V. DISCUSSION

One can reasonably hypothesize that maximizing capacity (the portion of the allowable message used by the transcoded

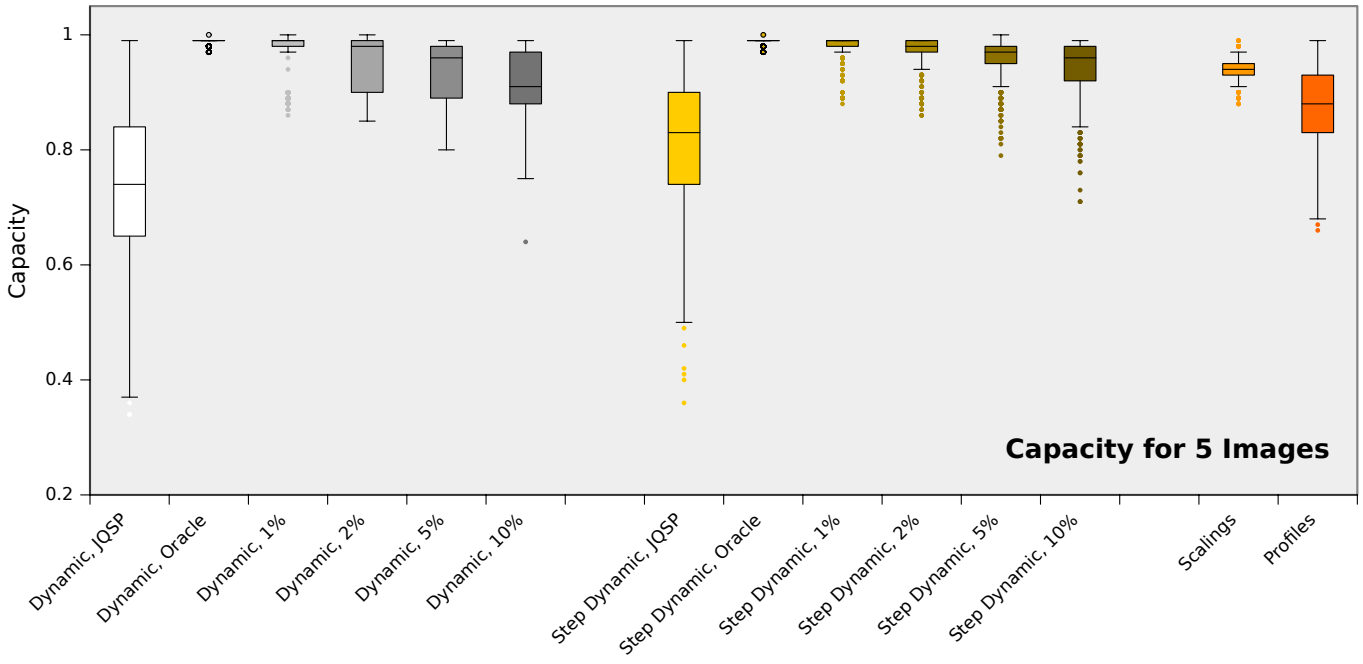

Fig. 1. Capacity resulting from the different algorithms, for 5 images.

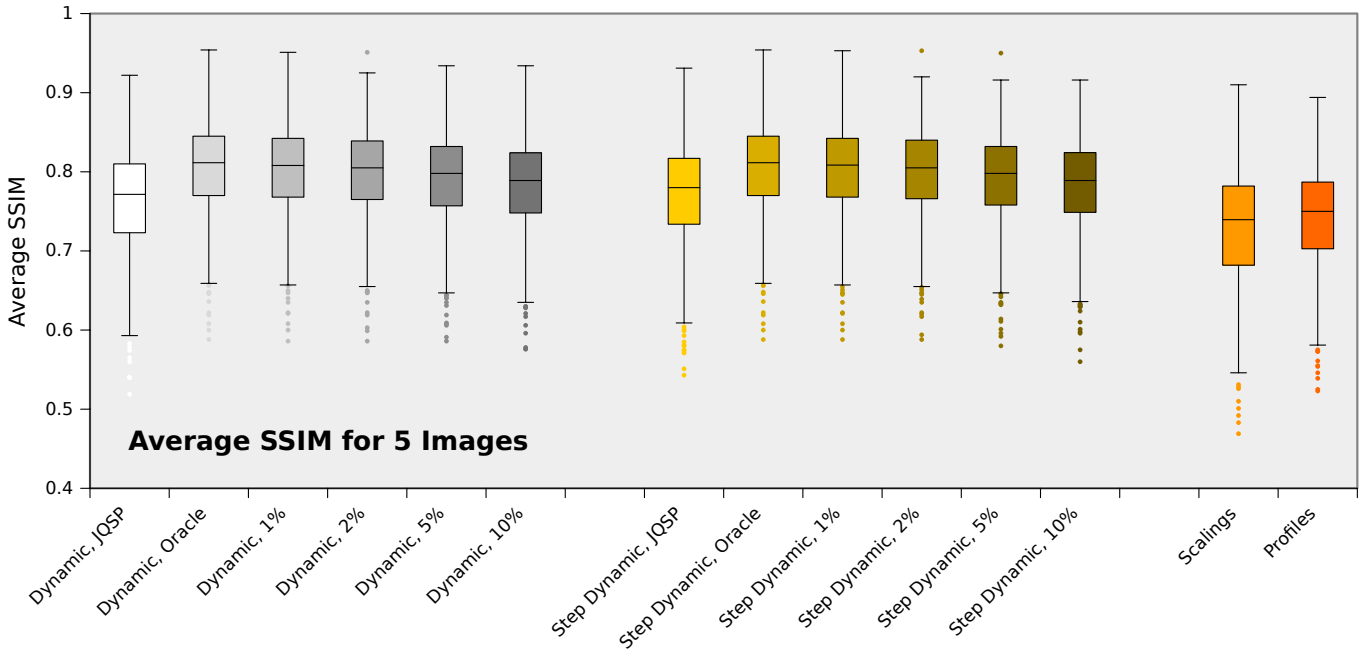

Fig. 2. Average SSIM resulting from the different algorithms, for 5 images.

TABLE II
SUMMARY FOR 2 ATTACHMENTS.

Optimization Algorithm	Predictor	Average Transcodings	Average Retries	Average Capacity	Average Quality	Objective Function
Dynamic Programming	Oracle	2.00	0.00	0.95	0.85	0.73
	±1%	2.04	0.03	0.93	0.85	0.72
	±2%	2.08	0.05	0.90	0.85	0.72
	±5%	2.16	0.08	0.84	0.84	0.71
	±10%	2.25	0.13	0.79	0.83	0.70
	JQSP	2.00	0.00	0.45	0.80	0.64
Step Dynamic Programming	Oracle	2.00	0.00	0.95	0.85	0.73
	±1%	2.02	0.01	0.93	0.85	0.72
	±2%	2.03	0.01	0.89	0.85	0.72
	±5%	2.04	0.02	0.81	0.84	0.71
	±10%	2.06	0.03	0.75	0.83	0.69
	JQSP	2.00	0.00	0.45	0.80	0.64
Scalings	—	4.55	1.28	0.93	0.82	0.67
Profiles	—	6.90	2.45	0.86	0.83	0.69

TABLE III
SUMMARY FOR 3 ATTACHMENTS.

Optimization Algorithm	Predictor	Average Transcodings	Average Retries	Average Capacity	Average Quality	Objective Function
Dynamic Programming	Oracle	3.00	0.00	0.98	0.84	0.59
	±1%	3.17	0.08	0.97	0.84	0.59
	±2%	3.32	0.13	0.95	0.83	0.58
	±5%	3.66	0.25	0.92	0.83	0.57
	±10%	3.89	0.32	0.88	0.82	0.55
	JQSP	3.01	0.01	0.61	0.80	0.51
Step Dynamic Programming	Oracle	3.00	0.00	0.98	0.84	0.59
	±1%	3.05	0.02	0.96	0.84	0.59
	±2%	3.06	0.02	0.95	0.83	0.58
	±5%	3.13	0.04	0.91	0.83	0.57
	±10%	3.11	0.04	0.87	0.82	0.55
	JQSP	3.02	0.01	0.63	0.80	0.51
Scalings	—	8.59	1.86	0.93	0.78	0.49
Profiles	—	14.83	3.94	0.86	0.78	0.48

TABLE IV
SUMMARY FOR 4 ATTACHMENTS.

Optimization Algorithm	Predictor	Average Transcodings	Average Retries	Average Capacity	Average Quality	Objective Function
Dynamic Programming	Oracle	4.00	0.00	0.99	0.83	0.47
	±1%	4.38	0.14	0.97	0.82	0.46
	±2%	4.62	0.21	0.96	0.82	0.46
	±5%	5.21	0.35	0.94	0.81	0.44
	±10%	5.88	0.51	0.90	0.80	0.42
	JQSP	4.03	0.03	0.70	0.78	0.38
Step Dynamic Programming	Oracle	4.00	0.00	0.99	0.83	0.47
	±1%	4.03	0.01	0.98	0.82	0.46
	±2%	4.09	0.02	0.97	0.82	0.46
	±5%	4.18	0.05	0.94	0.81	0.44
	±10%	4.30	0.08	0.92	0.80	0.42
	JQSP	4.12	0.03	0.75	0.79	0.39
Scalings	—	11.94	1.99	0.94	0.76	0.34
Profiles	—	23.62	4.91	0.84	0.76	0.34

TABLE V
SUMMARY FOR 5 ATTACHMENTS.

Optimization Algorithm	Predictor	Average Transcodings	Average Retries	Average Capacity	Average Quality	Objective Function
Dynamic Programming	Oracle	5.00	0.00	0.99	0.81	0.35
	±1%	5.57	0.17	0.97	0.80	0.34
	±2%	6.08	0.29	0.96	0.80	0.33
	±5%	6.93	0.45	0.94	0.79	0.32
	±10%	7.75	0.60	0.92	0.78	0.30
	JQSP	5.11	0.06	0.74	0.76	0.26
Step Dynamic Programming	Oracle	5.00	0.00	0.99	0.81	0.35
	±1%	5.04	0.01	0.98	0.80	0.34
	±2%	5.08	0.02	0.98	0.80	0.33
	±5%	5.22	0.05	0.96	0.79	0.32
	±10%	5.33	0.06	0.94	0.78	0.30
	JQSP	5.22	0.04	0.82	0.77	0.28
Scalings	—	15.02	2.00	0.94	0.73	0.23
Profiles	—	33.36	5.67	0.88	0.75	0.22

TABLE VI
TIMES, IN SECONDS

Number of Attachments	Dynamic Programming	Step Dynamic	Scalings	Profiles
2	0.09	0.09	0.19	0.31
3	0.13	0.12	0.33	0.62
4	0.18	0.17	0.47	0.99
5	0.21	0.20	0.55	1.30

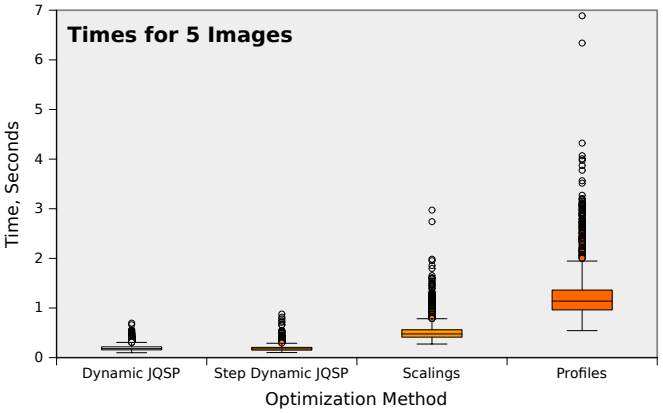

Fig. 3. Times, in seconds, for the different algorithms on 5 images.

images) is essentially equivalent to maximizing perceived quality, and *vice versa*. This is the hypothesis used by the heuristics of successive scalings and successive profiles that try merely to find the largest images (one considering arbitrary resolutions, the other considering only profiles-specific resolutions) that fits into the message in order to maximize quality. However, examining tables II to V, and figs. 1 and 2, we see that this hypothesis is not verified. While it is true that capacity and resulting message quality are correlated, we see that it does not suffice to maximize capacity to maximize quality. Indeed: the successive scalings adaptation method usually yield high capacity with resulting quality only comparing to the successive profiles method, but with much worse quality than the two proposed explicit optimization methods.

In a similar way, one could hypothesize that it suffice to maximize the average image quality as the objective function rather than an objective function such as eq. (1) (or eqs. (5) and (9)). Again, while for maximization, the average and the product of image quality are correlated, it is preferable to maximize the product as it has the distinct advantage of rejecting solutions where one or more of the transcoded images are of very poor quality, as maximizing the product (especially of values between 0 and 1) also requires maximizing individual image quality, with the side-effect of reducing variance [20]–[22]. Using an average or sum-like objective function, we could find ourselves with the case of a transcoding solution for a message with five images with four high-quality images but one image with exceedingly poor quality (which would be unacceptable) being chosen over a preferable solution where all five images are of approximatively equally good quality (therefore with a small variance), simply because the *average*

quality of the first solution is higher.

The quality of the predictor plays a major role in the quality of the transcoded messages, but the proposed algorithms will only degrade gracefully in the presence of increased predictor error. Examining fig. 1 for capacity and fig. 2 for resulting quality, we see that, indeed, the performance only degrades progressively as predictor error increases. With the error-free oracular predictor, both proposed algorithms, as expected, find very good solutions, using essentially all capacity yielding a high average quality solution (but due to the quantization of transcoding parameters, discussed in section III, the oracle may not find a solution using exactly 100% capacity). The JQSP predictor is doing much worse than the oracular predictors (as it is biased and overestimates file size [27]) but would compare to an oracular predictor with $\approx 15\%$ error.

Resilience to predictor error and bias is a major problem for optimization. If the predictor error is symmetric (and maybe vaguely gaussian), the errors would tend to cancel each other out; but if, like JQSP, the errors are asymmetric, an algorithm such as one-shot dynamic programming would not be able to cope with accumulated errors. However, the step dynamic programming solution can compensate for accumulated error as it transcodes one image, observes the transcoded size, and readjust its size constraint accordingly. Looking at fig. 1, it is clear that the by-step strategy allows the optimization algorithm to make much better use of the capacity (with lower variance) than the one-shot dynamic programming approach, an effect that is also seen on resulting average image quality, although to a lesser extent, as shown in fig. 2.

While absolute execution times of the algorithms are something of an implementation detail, it is nonetheless interesting to examine how implementations compare. Consider table VI and fig. 3. First let us compare dynamic programming versus step dynamic programming. Times show that, while both variants perform essentially the same number of transcodings, the execution times are comparable. This means that the optimization process is entirely dominated by the time for the actual transcoding, and that the time spent in the optimization *per se* is comparatively negligible. Figure 3 also reveals something noteworthy: the variance in execution time is much lower using the dynamic and step dynamic programming methods than with the comparative algorithms, successive scalings and successive profiles.

The number of transcoding parameters to examine will greatly influence optimization time as, even if the dynamic programming algorithm solving eq. (5) is computationally optimal, it is still essentially $O(nm^2)$, where n is the number of images and m the average number of transcoding parameters tested per image [24], [25]. As n is fixed (we are excluding the possibility of dropping images), a speed-up can only be gained by the reduction of m. The set of transcoding parameters series, whether the exact $\boldsymbol{T}(M, D)$ or the predicted $\widehat{\boldsymbol{T}}(M, D)$, can be pruned without affecting optimality by excluding transcodings yielding images exceeding either the maximum message size or maximum resolution for the receiving device. We can further reduce the complexity by considering prunings that affect the optimality of the solution. For example, one could exclude transcoding parameters that would yield very poor quality, defined by a user-specified threshold. One could also prune the set of possible transcodings to have transcoding parameters that yields (predicted) relative file sizes set at least 5% apart, or any other such heuristic that yields a satisfactory trade-off between parameter density, optimization speed, predictor error, retries, probability of failing to find a solution, and resulting adapted message quality.

In the same way, we could accelerate the successive profiles algorithm by considering even fewer profiles; which would make it faster, but also make it coarser and yield even worse results. One could be tempted, on the contrary, to *add* profiles. However, it would not be sufficient to merely add profiles, certainly not without changing how profiles are applied. The profiles are applied in the order shown in table I, stepping down resolution only when solutions using the lowest quality factors given the currently examined resolution have been explored. A better strategy would be to consider a profile, say, Image Rich, but with intermediate resolutions, such as 600×450 and 533×400 (other $4:3$ aspect ratio resolutions), in combination with different quality factors, but rather than trying a resolution with all its listed quality factors, then move on to the next resolution if no solution is found, it would be preferable to try combinations of resolution and quality factor in descending order of expected resulting file size. It would allow the successive profile algorithm to find solutions with smaller images encoded with a larger quality factor; although, obviously, it would not speed up the transcoding process.

For a high-volume service provider such as a telco operator, an algorithm that produces satisfactory adaptation of messages at the lowest possible computational cost (as adaptation rather mundanely translates into server racks, floor space, and electricity bills) is the preferable algorithm. In this work, we show that the proposed dynamic programming-based algorithms are interesting solutions, faring significantly better than comparative algorithms (both strongly inspired from what is found in commercial products and in previous literature). However, it is an open question as to how much sub-optimality—and how one defines optimality in this context—the users are willing to accept without taking notice of image/message degradation, and therefore which trade-off are available to service providers. One can surely consider using varying strategies depending on time of day and network traffic, possibly even adapting computational effort depending on the subscribers' data plans, network traffic, or other transient considerations. All these are user-experience considerations that cannot be addressed by the current work, but are certainly worthwhile exploring further.

VI. Conclusion

In this work, we have shown that the two proposed predictor-based dynamic programming multipart message adaptation algorithms maximize quality explicitly (as a proxy for user-experience), also making better use of message capacity (the portion of the allowable message size used) than the comparative algorithms inspired by actual prod-

ucts and in previous literature. We further show that while predictor accuracy is important, our proposed algorithms only degrade gracefully with predictor error increase, making them robust to prediction errors. We also show that the proposed algorithms are significantly faster and better than prior solutions.

REFERENCES

[1] Open Mobile Alliance, "Enabler Test Specification for (Conformance) for MMS Candidate Version 1.3," Oct. 2010, OMA-ETS-MMS_CON-V1_3-20101015-C.

[2] J. Dwyer, III, "MMS to Prosper as Mobile Marketing Becomes Mainstream," *Wireless Week*, Jan. 2011.

[3] S. Coulombe and G. Grassel, "Multimedia Adaptation for the Multimedia Messaging Service," *IEEE Communication Magazine*, vol. 42, no. 7, pp. 120–126, July 2004.

[4] Nokia, "How to Create MMS Services," Whitepaper, June 2003.

[5] S. Coulombe and S. Pigeon, "Low-Complexity Transcoding of JPEG Images with Near-Optimal Quality Using a Predictive Quality Factor and Scaling Parameters," *IEEE Trans. Image Processing*, vol. 19, no. 3, pp. 712–721, Mar. 2010.

[6] J. Ridge, "Efficient Transform-Domain Size and Resolution Reduction of Images," *Signal Processing: Image Communication*, vol. 18, no. 8, pp. 621–639, Sept. 2003.

[7] J. Mukherjee and S. K. Mitra, "Image Resizing in the Compressed Domain using Subband DCT," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 620–627, July 2002.

[8] D. Dugad and A. Ahuja, "A Fast Scheme for Image Size Change in the Compressed Domain," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 4, pp. 461–474, Apr. 2001.

[9] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas, "Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing," *IEEE Personal Communications Magazine*, vol. 5, no. 6, pp. 8–17, 1998.

[10] R. Mohan, J. R. Smith, and C.-S. Li, "Adapting Multimedia Internet Content for Universal Access," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 104–114, Mar. 1999.

[11] W.-Q. Yan and M. S. Kankanhalli, "Multimedia Simplification for Optimized MMS Synthesis," *ACM. Trans. Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 3, no. 1, Feb. 2007.

[12] R. C. S. Chen, S. J. H. Yang, and J. Zhang, "Enhancing The Precision of Content Analysis in Content Adaptation using Entropy-Based Fuzzy Reasoning," *Expert Systems with Applications*, vol. 37, pp. 5706–5719, 2010.

[13] S. Pigeon and S. 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.

[14] S. Coulombe and S. 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)*.

[15] S. Pigeon and S. Coulombe, "Optimal Quality-Aware Predictor-Based Adaptation of Multimedia Messages," in *Procs. The 6th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Sept. 2011, pp. 496–499.

[16] S. Rezazadeh and S. Coulombe, "A Novel Discrete Wavelet Domain Error-Based Image Quality Metric with Enhanced Perceptual Performance," *International Journal of Computer and Electrical Engineering*, vol. 4, no. 2, pp. 390–395, 2012.

[17] Z. Wang, A. C. Bovick, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[18] "The Independent JPEG Group," http://www.ijg.org/.

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

[20] T. 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*, vol. 12, no. 3, pp. 197–207, 2002.

[21] M. D. Springer and W. E. Thompson, "The Distribution of Products of Independent Random Variables," *SIAM Journal on Applied Mathematics*, vol. 14, no. 3, pp. 511–526, May 1966.

[22] M. D. Springer and W. E. Thompson, "The Distribution of the Products of Beta, Gamma and Gaussian Random Variables," *SIAM Journal on Applied Mathematics*, vol. 18, no. 4, pp. 721–737, June 1970.

[23] R. Bellman, *Dynamic Programming*, Dover, New York, 2003.

[24] F. S. Hilier and G. J. Lieberman, *Introduction to Operation Research*, McGraw-Hill Science, 9th edition, 2009.

[25] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover, 1998.

[26] "ImageMagick and Magick++," http://www.imagemagick.org/.

[27] S. Pigeon and S. Coulombe, "Efficient Clustering-based Algorithm for Predicting File Size and Structural Similarity of Transcoded JPEG Images," in *Procs. IEEE International Symposium on Multimedia (ISM)*, Dec. 2011, pp. 137–142.