Image Coding Based on Patch-Driven Inpainting

Nuno Couto[1,2], Matteo Naccari[2], Fernando Pereira[1,2]
Instituto Superior Técnico – Universidade de Lisboa[1], Instituto de Telecomunicações[2]
Lisboa, Portugal

***Abstract*— Recent advances in capturing and display technologies, as well as the proliferation of platforms to share images on the Internet, will further increase the bandwidth and storage space required by image coding based applications. To reduce the image coding rate, some techniques taking into account the properties of the human visual system can be used. In this context, this paper proposes an inpainting based image codec which is able to improve the image compression efficiency. The proposed codec builds on top of the JPEG image coding standard and its rate-distortion performance is assessed using a novel methodology, particularly suitable for perceptual image codecs as the one proposed in this paper. According to this methodology, the proposed image codec allows a bitrate reduction of up to about 20% regarding the JPEG standard, at the same perceptual quality**.

***Index Terms*—** *Image inpainting, patch-driven, JPEG coding, metric resolving power, objective quality assessment.*

1. Introduction

The increasing availability of high resolution image capturing devices at prices affordable for consumer users and the proliferation of image sharing platforms, such as Flickr, Panoramio, Picasa, etc., are continuously demanding for more bandwidth and storage capacity. This scenario calls for the design of new image coding algorithms to improve the compression efficiency beyond the JPEG [1] and JPEG 2000 [2] state-of-art image coding standards. This need is further supported by the JPEG Advanced Image Coding (AIC) project [3] which targets specifying a new image coding standard. One approach to improve this compression efficiency relies on the exploitation of the Human Visual Systems (HVS) properties by means of image inpainting [4]. Image inpainting is a rather well known image processing technology allowing to fill missing or damaged image areas, by imitating the procedures used by art experts and painting restorers. Although formerly proposed for image restoration, image inpainting can be also exploited in image coding by skipping at the encoder side the image blocks that may be inpainted with the target quality at the decoder. In this way, the coded bitrate for a given image is reduced since some areas are coded in this skipping mode, or only some features are sent, thus deeply exploiting the image spatial redundancy; if adequate skipped areas selection is performed, the distortion associated to the inpainted blocks in the decoded image may not be perceived by the HVS. The design of an inpainting based image codec requires addressing three main issues: 1) the encoder selection of the image blocks to be inpainted at the decoder; 2) the inpainting process to be used at the decoder; and, finally, 3) the appropriate assessment of the compression efficiency performance taking into account that this is a perceptually driven coding approach exploiting the HVS characteristics. In this context, the novelty brought by this paper is threefold: first, it proposes an inpainting based perceptual image coding architecture denoted as Instituto Superior Técnico-inpainting (*IST-inpainting*) codec; second, it proposes tools to address the two first aforementioned issues; and, third, it proposes an assessment methodology particularly suitable for perceptual image codecs since it takes into account the HVS varying distortion sensitivity as well as the nonlinear sensitivity of objective quality metrics observed especially when the perceived quality only slightly changes. While image coding is a rather well known topic, the truth is that *the impact of HVS features in the most popular image coding standards is unduly very limited*. This paper contributes to a new generation of image codecs which more deeply consider the visual perceptual characteristics. This is not a main stream approach but rather a less mature, alternative and promising approach, which is healthy for research.

The remainder of this paper is organized as follows: Section 2 reviews the very few inpainting based image codecs already proposed in the literature. Section 3 presents the proposed IST-inpainting architecture and details the encoder and decoder processes while Section 4 presents the proposed Rate-Distortion (RD) performance assessment methodology. Section 5 describes the experiments performed and the results obtained for the proposed IST-inpainting codec in comparison with a benchmarking JPEG codec. Finally, Section 6 concludes the paper and discusses future research directions.

2. Previous Work: a Brief Review

This section reviews the most relevant image coding architectures based on image inpainting already available in the literature. Given the central role that image inpainting plays in these codecs, the following review will start by briefly summarizing the seminal work on image inpainting [4]. In 2000, Bertalmio *et al.* [4] modeled the practical painting restorers' know-how as an image processing technique. For this technique, image inpainting consists in prolonging the lines of pixels with equal gray levels, called *isophotes*, inside the image region being inpainted. The algorithm iteratively fills the image area being inpainted according to the aforementioned isophotes prolongation. This approach leads sometimes to incorrectly reconstructed image

ISBN 978-0-9928626-7-1 © EURASIP 2017

areas [5]. In 2004, Criminisi *et al.* [5] observed that also the order by which each pixel is inpainted is important. To this end, the authors designed a filling order strategy based on the number of pixels available for inpainting in the neighborhood of the pixel being inpainted. This filling order concept is also exploited in the proposed IST-inpainting codec. While there are many image inpainting tools in the literature with restoration purposes, there are very few attempts to include such tools in image codecs. The use of inpainting tools for image coding has been firstly pursed in 2003 by Rane *et al.* [6]. In this work, image inpainting was used with the twofold goal of concealing the image blocks corrupted by channel noise as well as to improve the compression efficiency. Regarding the coding perspective, this work classifies each 8×8 image block which on a checkerboard pattern as texture or structure. Structure blocks are recovered at the decoder with the image inpainting algorithm in [4] while texture blocks are recovered with the texture synthesis method proposed in [7]; this later method replaces each missing pixel by searching, among the known pixels in the neighborhood, the one most similar. The main idea of the work in [6], i.e. encoder side block skipping and decoder side block inpainting, has been further extended in 2007 by Liu *et al.* [8]. This inpainting based image codec classifies each image block as *exemplar* or *non-exemplar*: the former blocks are coded with traditional coding methods, such as JPEG and H.264/AVC intra coding; for the non-exemplar blocks, only some side information (e.g. the block edges) is extracted and transmitted to the decoder side to guide their inpainting process. Its compression efficiency has been compared with the JPEG and the H.264/AVC standards with this latter operating only with Intra coding modes. Typically, the performance comparisons of inpainting based image codecs with relevant benchmarks, such as the JPEG, JPEG 2000 and H.264/AVC Intra codecs, are carried by measuring the rate savings for the same perceptual quality. While the image codecs described above mention some rate gains for the same perceptual quality, both the works in [6] and [8] assess these rate gains by comparing side-by-side images reconstructed with the inpainting based image codec and the adopted benchmark which is clearly a very limited and unreliable approach. To the best of the authors' knowledge, there are no attempts in the literature to measure these rate gains either by means of subjective testing or extensive results for well accepted objective quality metrics.

3. Proposed IST-Inpainting Codec

This section describes the proposed IST-inpainting codec and the details for both the encoder and the decoder coding tools. The overall architecture is depicted in Figure 1. The proposed IST-inpainting architecture combines a JPEG baseline codec with novel inpainting tools to be used for the image blocks inpainted at the decoder and, thus, not coded with a conventional, standard approach. In this paper, only the luminance image component is coded, as typically done in the literature, also because the proposed tools are straightforwardly extendable to the chrominance components.

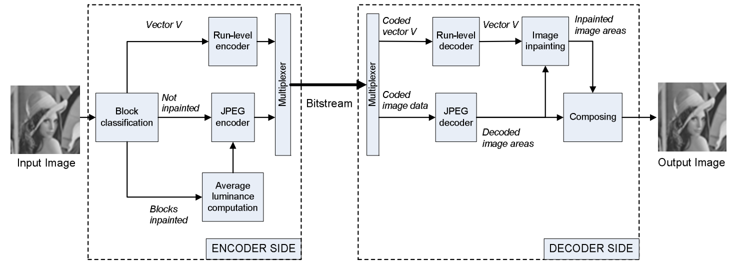

Figure 1: IST-inpainting codec architecture.

A. Encoding process

The overall encoding process for the luminance component Y of the input image I can be summarized by the following sequence of steps:

1. Split Y into non-overlapping 8×8 samples blocks
2. Create the vector V with length $(R/8)\times(C/8)$ where R and C denote the number of rows and columns, respectively
3. **Block classification**: For each 8×8 block, B, do:
    4. Classify B as a *structure* or *texture* block using the three classification levels described below.
    5. If B is *texture,* then set $V(B) = 1$ and set the corresponding samples in Y to the average luminance value μ of the samples in B, i.e. $Y(B) = \mu;$ otherwise, set $V(B) = 0.$
6. Go to Step *3* until all blocks are analyzed.
7. **Classification vector coding:** Perform run-length encoding of the values in vector V.
8. **Structural image coding:** Perform JPEG encoding of the image Y where only the average luminance is coded for the texture blocks.
9. **Bitstream creation:** Multiplex the coded image Y with the coded V run-lengths.

The classification in Step 4 divides the image blocks into structure blocks, which are typically more difficult to inpaint, and texture blocks for which inpainting with texture synthesis algorithms [7], usually provides a good reconstruction quality. The IST-inpainting codec performs decoder side inpainting only for texture blocks by means of a patch-driven approach; the choice for inpainting only this kind of blocks is mainly to limit the decoder complexity. The inclusion of the average luminance for the texture blocks in the JPEG coded image helps the differential coding of the DC coefficients as specified in the JPEG standard [1]. The novel block classification process proposed for the IST-inpainting codec is based on the following considerations:

a) Image blocks containing highly structural details are typically difficult to inpaint with only neighbor samples.
b) Image blocks being inpainted with high structural details in their neighborhood are difficult to be inpainted too.
c) Image blocks not containing highly structural details but which are quite different from their surrounding blocks are difficult to be inpainted too.

The concept of structurally detailed block is modeled by the percentage of connected edge samples contained in the block. The overall classification process performed by the

ISBN 978-0-9928626-7-1 © EURASIP 2017

proposed IST-inpainting codec is articulated in three following classification levels.

A.1 First block classification level

This classification level classifies image blocks into structure and texture according to their edge sample connectivity. The steps followed are:

1. Apply the Canny's edge detector 9 to *Y*.
2. For each 8×8 block *B*, measure its percentage of connected edge samples $P_C(B)$ as explained below.
3. Classify *B* as *texture,* if $P_C(B) < \alpha$ or as *structure* otherwise.
4. Provide the *texture* blocks to the second classification level.

In Step 2, an edge sample is defined as connected if at least one of its eight neighbors is an edge sample too. After exhaustive simulations, the threshold α has been set to 25%.

A.2 Second block classification level

This block classification level derives from the second general consideration stated above. In fact, it may happen that, although a block *B* has been classified as texture, it has in its neighborhood some structural blocks which typically lead to poor quality inpainting. Therefore, the IST-inpainting codec discards as texture those texture blocks with at least one structure block in their neighborhood:

$$B = \begin{cases} structure & \exists B' \in N_4(B) : B' \text{ is structure} \\ texture & otherwise \end{cases}, \quad (1)$$

where $N_4(B)$ denotes the four *B* neighboring blocks (i.e. top, bottom, left and right). The blocks classified as texture will be passed to the following and final classification level.

A.3 Third block classification level

The last block classification level derives from the third general consideration above. In fact, it may happen that a texture block, even if surrounded by other texture blocks, presents high variations in its sampless, such that the inpainting process will typically lead to a poor reconstruction quality. To avoid this situation, the following metric is computed for each texture block, *B*:

$$\Omega(B) = \sigma^2(B) + \sum_{J \in N_8} \left| \mu(J) - \mu(B) \right|, \quad (2)$$

where σ^2 denotes the variance of the samples in *B*, μ is the average sample value in *B* and N_8 denotes the set of eight *B* neighbors. The Ω metric quantifies how complex is to inpaint block *B* by taking into account both its *intra* complexity (σ^2 term) as well as its difference regarding the neighbors (sum term in equation (2)). Once Ω has been computed, each texture block *B* resulting from the second classification level can be classified as follows:

$$B = \begin{cases} structure & \Omega(B) > \tau \\ texture & \Omega(B) \le \tau \end{cases}. \quad (3)$$

To make the threshold τ automatically adaptive to the image content, a novel threshold computation method is proposed. The τ value is then computed as the centroid of the normalized histogram of the Ω values for all the blocks analyzed in this classification level. Finally, not all the texture blocks classified at this level will be inpainted at the decoder but only those whose position lies on a checkerboard pattern defined within the texture areas. This final step targets avoiding to inpaint large textures areas since this would lead to a poor inpainting quality of the innermost blocks.

The three above proposed block classification levels are illustrated for the *Lena* image in Figure 2. Figure 2(a) represents the first level with the blue blocks corresponding to the texture blocks while the gray blocks correspond to the structural ones. Figure 2(b) shows the results of the second classification level with the pink blocks corresponding to the texture blocks with structural blocks in their neighborhood and the green blocks corresponding to the textural blocks that will undergo the third classification level. Finally, Figure 2(c) shows the results of the final classification level, including the block selection according to the aforementioned checkerboard pattern in orange.

Figure 2: Results for the three classification levels for *Lena*.

B. Decoding process

At the decoder side, the 8×8 image blocks classified as structure by the encoder are JPEG decoded; in this process also the block average luminance for the texture blocks is obtained. Conversely, the texture blocks are to be inpainted by means of a patch-driven texture synthesis approach [7]. In this context, a patch is a square area of 3×3 samples centered at the image sample being inpainted. Once the inpainting process is finished, the IST-inpainting codec performs a block average luminance adjustment, exploiting the average luminance values available for texture blocks. The overall decoding process can be summarized by the following sequence of steps:

1. **Classification vector decoding**: Perform run-length decoding of the *coded vector V,* thus obtaining the texture/structure classification for each block.
2. **Structural image decoding**: Perform JPEG decoding of the blocks whose entry in *V* is marked as *structure*. Mark all the samples in *texture* blocks as *to be inpainted*.
3. **Inpainting process**: while all the samples to be inpainted have not been inpainted do
    4. Compute the priority (see below) for the patch corresponding to each sample to be inpainted.
    5. Select the sample to be inpainted *p* whose patch has the highest priority.
    6. Perform inpainting for sample *p* and mark *p* as inpainted.
7. Go to Step 3.

ISBN 978-0-9928626-7-1

8. **Average luminance correction**: For all the samples in *texture* blocks, add the difference between the decoded average luminance value and the average luminance value of the inpainted samples.

Steps 4 and 6 represent the core of the inpainting based decoding process and will be detailed in the following.

B.1 Patch priority computation

According to [5], the sample inpainting order is very important for the final obtained quality. In fact, there are samples to be inpainted whose associated patches contain a high number of know sample values, i.e. either JPEG decoded samples or already inpainted samples. These patches should receive higher priority with respect to those containing a fewer number of known sample values as they provide, in principle, more reliable texture information. To compute the priority for the patch associated to the sample being inpainted, the approach suggested in [5] has been adopted. Therefore, let $\Pi(p)$ be the patch associated to the sample p being inpainted; the patch priority for $\Pi(p)$, $P(\Pi(p))$, is given by:

$$P(\Pi(p)) = \left(\sum_{q \in \Gamma_\Pi} C(q) \right) / |\Pi(p)|, \qquad (4)$$

where $|\Pi(p)|$ denotes the $\Pi(p)$ area, Γ_Π the set of samples belonging to $\Pi(p)$ whose value is known and $C(q)$ the confidence associated to sample $q \neq p$. This confidence expresses how much a patch is suitable for being used in the inpainting process. At the beginning of the inpainting process, the confidence for the samples being inpainted is set to zero whereas the confidence for the JPEG decoded samples is set to one. After a sample p has been inpainted, its confidence becomes $C(p) = P(\Pi(p))$.

B.2 Sample inpainting

To determine the inpainted sample value, this work considers the patch Π centered in the sample p being inpainted and looks, in a search range of size S, for the patch Π^* minimizing the distance d between Π and Π^* [7]. Once Π^* has been found, the sample p is inpainted with the value of sample q at the center of Π^*. In this paper, the size S has been set to 11 samples whereas the distance d has been measured in terms of Sum of Square Difference (SSD).

4. Proposed Assessment Methodology

Since the proposed codec exploits the HVS properties to allow efficiently masking the inpainting distortion, it is not possible to assess its benefits using the most traditional image quality metric (i.e. Peak Signal-to-Noise Ratio (PSNR)). The proposed RD performance methodology provides a relative assessment as it evaluates the performance of one image codec taking another codec as reference. Moreover, the proposed assessment methodology considers the nonlinear sensitivity of objective metrics, observed especially when the perceptual quality changes slightly. To further clarify this latter claim, consider the example of two image codecs (A, B) encoding the same image with codec A providing the reference quality and codec B providing a subjective quality which is just noticeable different. Although the subjective quality is thus basically the same, it may happen that the objective quality scores are significantly different leading to the conclusion that the quality of codec B is drastically different from A even if the adopted metric is known to generally correlate well with the perceived quality. To avoid this limitation, the proposed assessment methodology uses the Metric Resolving Power (MRP) concept [10]. The MRP corresponds to the maximum change in the adopted Quality Metric (QM) scores for which the perceived quality can be considered the same [10]. In other words, if the absolute difference between the objective scores of codecs A and B is less than the MRP, then their quality will be perceived as similar by a human observer. Therefore, the objective score of B can be compensated to the one of codec A. In this context, the objective score compensation is done by setting the objective score of codec B equal to the one of the reference codec A. More specifically, the proposed assessment methodology consists in the following steps:

1. Compute the resolving power for the chosen objective *QM* as specified in 10 and set it to *MRP*.
2. For each coded test image *I* do:
    3. Compute the *QM* for codec A and set it to $QM_A(I)$.
    4. Compute the *QM* for codec B and set it to $QM_B(I)$.
    5. If $|QM_A(I)–QM_B(I)|<MRP$ then set $QM_B(I)= QM_A(I)$.
6. Go to next image *I*.

The condition $|QM_A(I)–QM_B(I)|<MRP$ is denoted as Resolving Power Equivalence (RPE) condition and determines when the two objective quality scores are close enough for the subjective quality to be equivalent.

5. Experimental Assessment

This section presents the RD performance of the IST-inpainting codec. To this end, four images, namely *Lena*, *Peppers*, *Jet* and *Mandrill* (512×512 Y resolution) have been used [11]. These images span a wide range of texture and structure details and have been JPEG encoded at six bpp rates, notably 0.25, 0.5, 0.75, 1.00, 1.25 and 1.5 bpp, using the JPEG parameters in [12]. According to the proposed assessment methodology, JPEG is taken as the reference codec. Then, the IST-inpainting bitrate is compared with the JPEG one for the same perceptual quality level. In this paper, the perceptual quality has been objectively quantified by means of the Structural SIMilarity (SSIM) quality metric [13] due to its good Pearson's correlation (up to 0.97) with human perceived image quality [13]. The SSIM scores for both the JPEG and the IST-inpainting codecs have been analyzed according to the assessment methodology proposed in Section 4. The SSIM resolving power has been computed according to the procedure presented in [10], using the test images and the subjective data collected in a recent study [12]. The obtained SSIM resolving power was 0.09 given that SSIM has quality scores in the range [0,1], with 1 corresponding to excellent quality compared to the original. Table 1 reports the measured RD performance over the tested images for both the JPEG and IST-inpainting codecs. The reported SSIM scores are valid for both codecs as the RPE condition was always satisfied for all the tested images and rates. In this scenario, the rate reductions can be measured as reported in the last column of Table 1.

From the results, some general conclusions can be drawn: first, a rate reduction of up to -18% regarding JPEG can be achieved; second, the IST-inpainting bitrate reductions show an increasing trend with the bitrate. This trend is expectable as the rate difference between the JPEG and IST-inpainting codecs is associated to the inpainted image blocks whose rate becomes significant at higher bitrates; third, images with many smooth areas, such as *Lena* and *Peppers,* provide the highest bitrate reductions as the "*Block classification*" module will select more blocks to be inpainted. At very low rates, the IST-inpainting codec may perform worse than JPEG due to the rate for coding the classification vector, *V*. In fact, for very low rates, the IST-inpainting codec reductions regarding JPEG are rather small, thus not compensating the bitrate associated to *V*. However, it may be argued that more efficient entropy coding methods can be used to code *V*. A couple of visual comparison examples are presented in Figure 3 and Figure 4.

TABLE 1: IST-INPAINTING VERSUS JPEG RD PERFORMANCE COMPARISON.

Image	JPEG SSIM	JPEG rate [bpp]	IST-inpainting rate [bpp]	Δ rate [%]
Lena 	0.809	0.247	0.250	1.21
	0.905	0.495	0.467	-5.66
	0.931	0.745	0.671	-9.93
	0.944	0.993	0.871	-12.29
	0.952	1.237	1.059	-14.39
	0.958	1.442	1.216	-15.67
Peppers 	0.770	0.243	0.238	-2.06
	0.869	0.489	0.438	-10.43
	0.895	0.744	0.635	-14.65
	0.910	0.990	0.868	-12.32
	0.919	1.215	1.002	-17.53
	0.927	1.484	1.208	-18.60
Jet 	0.798	0.231	0.235	1.73
	0.916	0.495	0.474	-4.24
	0.947	0.745	0.686	-7.92
	0.960	0.997	0.887	-11.03
	0.967	1.214	1.063	-12.44
	0.972	1.451	1.245	-14.20
Mandrill 	0.450	0.214	0.213	-0.47
	0.682	0.474	0.463	-2.32
	0.769	0.719	0.691	-3.89
	0.822	0.977	0.923	-5.53
	0.857	1.229	1.147	-6.67
	0.884	1.490	1.377	-7.58
Average	-	-	-	-8.62

6. CONCLUSIONS AND FUTURE WORK

This paper has proposed an inpainting based image codec built on top of a JPEG codec. The RD performance has been assessed by means of a novel methodology taking into account the perceptual nature of inpainting techniques as well as the nonlinear sensitivity of objective metrics when the perceptual quality slightly varies. According to this methodology, the proposed codec allows a rate reduction of up to 18% regarding the JPEG standard, at the same perceptual quality. Furthermore, the IST-inpainting decoder is less complex than those proposed in [6] and [8]. Future research directions are the integration of inpainting tools in the HEVC video coding standard and the design of new inpainting techniques for the structure blocks.

Figure 3: *Lena*: (left) JPEG coded at 0.993 bpp and (right) IST-inpainting coded at 0.871 bpp.

Figure 4: *Peppers*: (left) JPEG coded at 0.744 bpp and (right) IST-inpainting coded at 0.635 bpp.

REFERENCES

1. ISO/IEC, "Digital compression and coding of continuous tone still images: requirements and guidelines", ITU Rec. T.81, September 1992.
2. S. Taubman and M. W. Marcellin, "Image compression fundamentals, standards and practice: JPEG2000", Kluwer, pages 773, 2002.
3. ISO/IEC, "Information technology - Advanced image coding and evaluation methodologies", JTC 1/SC 29/WG 1, 2017.
4. M. Bertalmio *et al.,* "Image inpanting", in Proc. of ACM Int. Conf and Exhibition on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, Jul. 2000.
5. A. Criminisi *et al.,* "Region filling and object removal by exemplar-based image inpainting", IEEE T-IP, vol. 13, no. 9, pp. 1200-1212, Sep. 2004.
6. S. D. Rane *et al.*, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications", IEEE T-IP, vol. 12, no. 3, pp. 296-303, Mar. 2003.
7. A. Efros and T. K. Leung, "Texture synthesis by nonparametric sampling", in Proceedings of the IEEE Int. Conf. on Computer Vision, Corfu, GR, Sep. 1999.
8. D. Liu *et al.,* "Image compression with edge based inpainting", IEEE T-CSVT, vol. 17, no. 10, pp. 1273-1287, Oct. 2007.
9. J. Canny, "A computational approach to edge detection", IEEE T-PAMI, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986.
10. M. H. Brill *et al.,* "Accuracy and cross-calibration of video quality metrics: new methods from ATIS/T1A1", Signal Processing: Image Communication, vol. 19, no. 2, pp. 101-107, Feb. 2004.
11. "The USC-SIPI image database", available at [http://sipi.usc.edu/database/]
12. F. de Simone *et al.,* "Subjective evaluation of JPEG XR image compression", in Proc. of SPIE Optics and Photonics, Applications of Digital Image Processing XXXII, San Diego, CA, USA, Aug. 2009.
13. Z. Wang *et al.,* "Image quality assessment: from error visibility to structural similarity," IEEE T-IP, vol.13, no.4, pp. 600-612, Apr. 2004.

ISBN 978-0-9928626-7-1