2015 11th International Conference on Signal-Image Technology & Internet-Based Systems

Exploiting Change Blindness for Image Compression

Steven Le Moan, Ivar Farup
Gjøvik University College, Norway
steven.lemoan@gmail.com, ivar.farup@hig.no

***Abstract*—Change blindness refers to the inability of our visual system to memorize details in pictures. We suggest that, under certain conditions, regions containing such details can be identified and altered in a way that benefits image compression. We define *change blindness maps* and a compliant texture synthesis method to identify, remove and subsequently recover pixels which are subject to change blindness. Unlike comparable methods, ours is particularly intended for images with complex textures. We demonstrate via a subjective experiment that up to 15% of such an image can be altered in a way that is considered as natural and acceptable.**

***Keywords*-perception; attention; saliency; rendering ; data reduction ; inpainting**

I. Introduction

Traditionally, in image processing and computer vision applications, models of human perception are used to assess which information - in an image - can be conveyed by our visual system and which cannot. This is used for example to predict subjective judgment of quality. The common "full-reference"' image quality assessment framework assumes that observers have access to both stimuli to compare at a same time, so that they can rate the perceived difference between them [27]. If these stimuli are shown sequentially however, one needs to account also for the influence of memory and internal representations. Several studies have indeed suggested that our representations of visual scene are sparse, incomplete or even nonexistent [17], but there is no clear evidence to validate either of these hypothesis yet. We can however infer from a phenomenon called *change blindness* [21] that, to a certain extent, our representations and the way we access them simply do not contribute to change perception [14].

In this paper, we would like to suggest that change blindness can be exploited for image processing and in particular for data reduction. We argue that, in any greyscale or color image, there are some textured regions that can be somewhat altered without immediately perceivable change and that these regions are those having both a low saliency and a high resemblance with the rest of the image. In order to automatically identify these regions, we propose an algorithm based on saliency detection and texture synthesis (alternatively referred to as *exemplar-based inpainting*) to create a confidence-adjusted saliency map, which we refer to as *change blindness map*. Pixels with low energy on the map can simply be removed on the encoder's side, with high confidence that they can be recovered from the remaining pixels on the decoder's side with minimal perceivable change. In order to make the method compliant with block-based encoding methods such as JPEG, we process pixels by macro-blocks (typically 8x8 square), not individually. Note that we do not discuss encoding or decoding strategies as our method is only intended to reduce the amount of data to encode, and has to be used together with an encoding/decoding scheme.

The method is particularly intended for pictures with complex textured background, as the one shown in Figure 1. The restored image, while different from the original one, can be considered as visually equivalent and we demonstrate via a user study that the discrepancies between the two are mostly acceptable.

II. Related work

Digital images captured by modern cameras contain plenty of visual data, to the point that most of it is not even perceptible in typical viewing conditions. Identifying the threshold between what is visible and what is not is of particular interest for efficient data reduction. For example, the popular JPEG or JPEG2000 formats are based on the fact that very high frequencies components can be represented coarsely without *disturbance* for the user [22].

Several studies also suggested to exploit spatial redundancies in textures for image compression. Rane *et al.* [20] proposed a basic scheme to delete macro-blocks in an image, which can be correctly reconstructed by partial difference equation-based inpainting (for structured regions) or texture synthesis (for textured regions). Their method uses a simple coarseness measure based on the number of local extrema to decide whether a region is structured or textured, which makes it particularly dedicated for simple and repetitive textures. Liu *et al.* [13] and later Xiong *et al.* [25] proposed to extract small-sized descriptors of the edge or gradient content of the blocks to help their recovery on the decoder's side. It allows to drastically reduce the amount of data to encode while preserving the perceived quality of the scene, yet it is mostly suitable for images with large uniform non-textured regions. This kind of approach which consists of using block descriptors extracted on the encoder's side to assist the synthesis on the decoder's side has been popular also in video compression [10], [19]. The fact that video frames are visible for only a fraction of a second makes it convenient to conceal synthesis artifacts in the least salient regions [23] and consequently allows to remove more blocks than if they were still images.

978-1-4673-9721-6/15 $31.00 © 2015 IEEE
DOI 10.1109/SITIS.2015.20

IEEE computer society

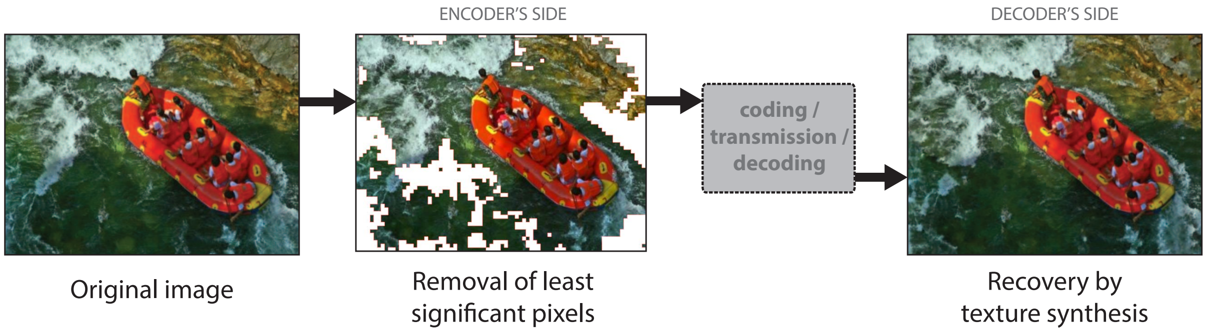

Figure 1. Workflow of the proposed method to assist image coding. The least significant pixels are those with a low saliency and a high probability of being restored accurately. The result, though perceptibly different from the original, suffices to satisfy the user in his viewing experience.

What is common to all these previous efforts is that the changes made to the data are meant to be imperceptible. Additionally, they rely on the extraction of simple block descriptors to help during the synthesis (except [20]). Therefore, they are not adapted for the compression of still images with complex and irregular textures which cannot be described with only a small number of attributes. One solution is to allow a perceptible difference, so long as the result remains visually appealing. Hays *et al.* [7] argued that it is not always necessary to restore an image the way it *should* have been to make it pleasing, but instead it is possible to render it the way it *could* have been. They proposed an image completion strategy that fills in large missing regions in an image with matching regions from other images. The results, though not real *per se*, are appealing.

III. Contributions

Our study is inspired by Hays' paper in that we believe that the semantic content of an image can be modified intentionally in a manner that does not necessarily disturb the observer. We argue that if a change in a scene cannot be located rapidly it can be considered as not disturbing the viewing experience[1], thus implying that the original and changed images are visually equivalent.

We propose a new approach to data reduction based on saliency detection and texture synthesis. Unlike related work, it allows the resulting image to be noticeably different from the original, while not disturbingly so. On the encoder's side, regions with low saliency and which are potentially easy to recover are completely discarded, thus reducing the amount of data to encode. On the decoder's side, texture synthesis is used to recover the missing pixels by generating texture composites from the known regions. Intentionally removing pixels from an image generates strong discontinuities on the border between known and unknown regions, which are difficult to encode efficiently. Many formats such as JPEG decompose the image in small blocks and compress them individually, which suggests an easy way around the discontinuity problem: erase entire blocks instead of pixels so that the discontinuities are invisible to the encoder. The resulting images after synthesis, though perceptibly different from the original ones, are visually appealing. Furthermore, we demonstrate with two subjective experiments that the changes made to the images do not interfere significantly with the user's viewing experience. Figure 1 illustrates the workflow.

To our knowledge, this is the first attempt at exploiting change blindness for still image compression. This is particularly important as we aim at making changes which *can* be noticed, unlike in the previously cited studies. Our contributions are as follow:

- We demonstrate that change blindness can - to some extent - be exploited for data reduction in digital pictures.
- We define so-called *change blindness maps* and propose a simple approach to compute them. Unlike comparable approaches, ours relies on saliency, does not use any block or texture descriptor and is particularly intended for pictures containing complex textures.
- We propose to use saliency to reduce the patch search space for inpainting.
- We introduce results from two subjective experiments which are freely available at http://www.colourlab.no/cid.

IV. Proposed approach

A. Change blindness maps

In an image $\mathbf{I}$, let us note $\mathbf{\Lambda_I}$ the set of pixels which could potentially be modified without disturbing the observer's viewing experience. As previously mentioned, attention seems a reasonable criterion to find $\mathbf{\Lambda_I}$. It is well-known that visual attention is influenced by a variety of top-down mechanism such as task, memory or culture, which are not yet well understood, however we can rely on the proven efficiency of some bottom-up methods such as the Boolean Map-based Saliency (BMS) detector by Zhang *et al.* [26], which have been reported to give among the best predictions of eye fixations on images from various databases. For our experiments, we manually

[1] By *viewing experience*, we refer generally to the interpretation of a picture, i.e. the conversion from a visual stimulus to some kind of semantic information.

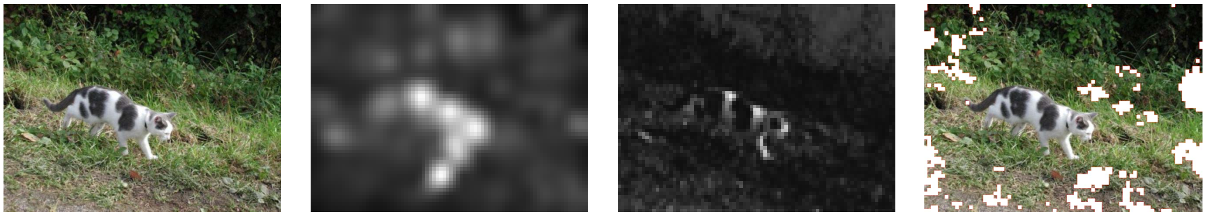

Figure 2. From left to right: original scene (originally 640x480 pixels), boolean map-based saliency map $\mathbf{S_I}$ [26], inverse inpaintability map $1-\mathbf{\Gamma_I}$ (dark locations indicate high inpaintability) according to the measure of patch difference described in Equation (5) and the result of block removal based on $\mathbf{CB_I}$ for $\rho = 10\%$. Note that the maps were both resized for this illustration, their original size is 80x60: each pixel in the original maps corresponds to a block in the picture.

selected a subset of images from one of these databases: ImgSal [11]. Even though there exist a substantial amount of other saliency detectors, we do not discuss in details the pros and cons of using either of them (for this, refer to [2]), instead we rely on data that we know to be accurate in order to validate our hypothesis.

Nevertheless, saliency may not always be a sufficient criterion to find $\mathbf{\Lambda_I}$ as the least salient pixels are not necessarily the easiest ones to replace. What is needed is a good compromise between low saliency and high *inpaintability*[2]. Therefore we propose to use a simple combination of a saliency map $\mathbf{S_I}$ (obtained with the BMS model) and an inpaintability map $\mathbf{\Gamma_I}$.

Let $\mathbf{\Psi_I}$ be the set of non-overlapping blocks into which $\mathbf{I}$ is segmented[3] and let us consider $d(\psi_i;\psi_j)$ a measure of the difference between any two blocks ψ_i and ψ_j. Also, let $o(\psi_i;\psi_j)$ be the offset (spatial distance) between the two patches' centers. We define the inpaintability of $\psi_i \in \mathbf{\Psi_I}$ as the inverse average of the product of these terms:

$$\gamma(\psi_i) = \frac{|\mathbf{\Psi_I}| - 1}{\sum\limits_{\psi_j \in \mathbf{\Psi_I}; j \neq i} d(\psi_i;\psi_j) o(\psi_i;\psi_j) + \epsilon} \qquad (1)$$

where $|\mathbf{\Psi_I}|$ represents the cardinality of $\mathbf{\Psi_I}$. The offset term favorizes the search for similar pixels in a local neighborhood. It has two beneficial effects: first, it makes $\gamma(\psi_i)$ account for local saliency, therefore giving lower inpaintability to textured regions that stand out from their neighborhood, regardless how well they could be reconstructed from the remaining pixels. We found that this seems to improve the accuracy of the change blindness map. Secondly, it is compliant with our exemplar search strategy (see Section IV-B1), which relies on saliency and proximity. According to this equation, the blocks with the highest inpaintability are those which have similar immediate neighbors. The measure d must be the same one used later for texture synthesis (see Section IV-B2). The infinitesimal term ϵ is intended to deal with the case in which $d(\psi_i;\psi_j) = 0, \forall\,(i,j)$ (case of a uniform image). Note that this measure is similar to the one used to detect local saliency in [1].

[2]We define the *inpaintability* of a set of pixels as the probability that it can be replaced in a visually appealing manner, with a given inpainting method.

[3]Note that if the image's dimensions are not dividable by the block size, a simple padding can be used.

We then propose to define the *change blindness map* $\mathbf{CB_I}$ as:

$$\mathbf{CB_I} = (\mathbf{S_I} + \mathbf{\Gamma_I})/2 \qquad (2)$$

where $\mathbf{\Gamma_I}$ is the inpaintability map depicting the spatial arrangement of $\gamma(\psi_i)$ for $\psi_i \in \mathbf{I}$, scaled to be in the range $[0..1]$. Note that $\mathbf{S_I}$ and $\mathbf{\Gamma_I}$ should be computed (or re-sized to be) at the same scale. Figure 2 shows an example. The number of pixels to be removed is a user-defined parameter which we note ρ. Eventually, $\mathbf{\Lambda_I}$ is identified as the set of ρ pixels with the lowest energy on $\mathbf{CB_I}$.

Note that the higher the resolution and/or the sharpness of the saliency map, the higher the chance that $\mathbf{\Lambda_I}$ is spatially sparse on $\mathbf{I}$ (i.e. only small neighborhoods are removed, which makes the texture synthesis easier because better guided).

B. Recovering the dropped pixels

On the decoder's side, the missing blocks are recovered by exemplar-based texture synthesis. This means that visually meaningful content is generated from known pixels, and pasted seamlessly in the unknown regions. There is a vast literature on image completion and inpainting [6], and we do not claim that the method described in this section surpasses the state-of-the-art in terms of speed or quality. We do however suggest that it is compliant with the way $\mathbf{CB_I}$ is computed and used. Recall that this study is primarily intended to demonstrate that change blindness can be exploited for image processing. Other inpainting strategies might perform better at seamlessly altering $\mathbf{\Lambda_I}$ on some images, yet our method is functional, inspired by the state-of-the-art in patch-based inpainting and allows to prove our hypothesis. Note finally that all the processes described here were made in the hue-linearized LAB2000HL color space [12], which exhibits more perceptual uniformity than CIELAB overall.

1) Patch dictionary: In order to recover missing pixels, the common strategy used in patch-based texture synthesis is to create a dictionary of complete patches from the known regions of the image, which is then used as exemplar to synthesize the missing pixels. It is well-known that the size of the patches is a critical parameter that will strongly influence the quality of the synthesis [6]. Small patches increase the size of the dictionary and therefore the probability of a good patch matching, but reduce the ability to deal with large textures. Large patches on the other hand reduce the chance of blurring artifacts or repeating structures, but allow to generate less varied content. Note that, in our experiments, patches were allowed to overlap with each other, rotate, and flip, so as to increase the size of the dictionary.

However in any case, it is possible to identify subsets of patches from the dictionary which have a higher probability to contain a good match [8]. We propose a new strategy based on saliency. We consider three levels of saliency in an image: the foreground (most salient pixels), the background (least salient pixels), and a level in-between which we refer to as middleground. Equation 2 implies that the dropped patches belong mostly to the background, and consequently have little resemblance with the foreground. Therefore the patch dictionary to reconstruct $\mathbf{\Lambda_I}$ does not need to contain patches from the foreground. Following this, we first isolate the foreground and build up the said dictionary with the middleground only (see Figure 3). For the foreground extraction, we used the method presented in [26]. Note that, because the saliency map cannot be recovered from the kept patches only, this implies that the foreground extraction should be done on the encoder's side, and that each kept block should be encoded and transmitted with a one bit-long overhead indicating whether it belongs to the foreground or not.

2) Patch matching: A typical difference/distance measure for nearest neighbor search in the patch domain is the sum of squared distances (SSD), i.e., the sum of all pixel-wise color differences [5]. It is simple, but limited in terms of accuracy as it does not fully capture the perceived differences between patches. Additionally, Bugeau *et al.* [4] observed indeed that the SSD, when used alone, tends to favor uniform patches. In order to improve this, we suggest to use the SSIM index' [24] contrast term. The SSIM index is a popular so-called image quality metric[4], which was reported to be a good predictor of human judgment. It defines the contrast similarity between two image patches ψ_i and ψ_j as:

$$c(\psi_i; \psi_j) = \frac{2\sigma_i\sigma_j + B}{\sigma_i^2 + \sigma_j^2 + B} \tag{3}$$

where σ_i and σ_j are the estimated standard deviations of lightness values in patches ψ_i and ψ_j, respectively, and B is a normalization parameter. This term ensures that patches are compared in terms of their standard deviations, thus avoiding a bias towards uniform patches.

The resulting measure of the difference between patches is:

$$d(\psi_i; \psi_j) = \frac{ssd(\psi_i; \psi_j)}{c(\psi_i; \psi_j)} \tag{4}$$

where $ssd(\psi_i; \psi_j)$ is the aforementioned sum of squared distances between pixels in patches ψ_i and ψ_j.

Note that this measure is the one that should be used to compute the inpaintability map $\mathbf{\Gamma_I}$ in order to make our workflow consistent. Although the compression/decompression process may modify the content of the image, we believe that the changes are insignificant to d, especially in high frequencies. This is particularly important to ensure that the exact same measure of inpaintability is used on both the encoder's and decoder's sides. In our experiments, we considered that the encoding/transmission/decoding unit does not alter the perceived quality of the blocks which are kept.

3) Processing order: The unknown regions are filled from the outside in. Incomplete patches are sampled randomly in $\mathbf{I}$ and only patches with at least 50% of known pixels are considered for matching and filling.

4) Stitching: Stitching patches together helps giving a sense of continuity between the recovered regions and the rest of the image. First, the optimal seam between patches is found via graphcut [9], then the overlap is fused in the gradient domain via Poisson blending [18].

V. Implementation

A. Parameters

As previously mentioned, the parameter with the most important influence on the appearance of the results is the patch size. In our experiments, we computed renderings for four square patch sizes: 15, 25, 35 and 45 pixels. All values produced visually pleasing results, but with different levels of texture coherence and semantic consistence. We found that state-of-the-art no-reference image quality indices such as BRISQUE [15] and inpainting quality indices such as BorSal [16] perform poorly at selecting what we believe were the best renderings, therefore we did the selection manually.

Parameter ρ defines the proportion of patches to be removed, and consequently the size of the exemplar (remaining patches). On images with textured background and spatially sparse foreground such as those shown in Figure 4, we observed that our method works best for $\rho \leq 15\%$, but this depends of course on the nature and variety of textures in the scene. Larger values result in both a smaller dictionary and larger regions to be filled and consequently in a lower quality of inpainting.

B. Computational efficiency

As an indication of the computational efficiency of our texture synthesis strategy, it takes about four minutes for a 700x500 image with 10% of missing pixels, with a patch size of 25x25 and with an unoptimized Matlab implementation. Because of this, our scheme is particularly intended for applications in which a fast transmission or a small file size are more critical requirements than decoding time.

VI. Experimental Validation

We tested our method on 20 natural images from the ImgSal database [11], selected for their textured content. All images were 640x480 pixels of size with three color channels. Figure 4 shows some of the most convincing renderings. We carried out two subjective experiments in

[4]Note that the term *metric* is not used in its proper mathematical definition. It is however commonly used in the image quality literature.

Figure 3. From left to right: original scene, blocks of low energy on $\mathbf{CB_I}$, foreground blocks (isolated by thresholding the saliency map [26]) and the remainder, used to build the exemplar dictionary.

order to test our hypothesis that change the original and restored images can be considered as visually equivalent.

A. First experiment

In a first study, 15 color-normal observers were asked to spot differences between the original and reproduced image shown one after the other. First an original image was displayed for a duration of 5 seconds, then a 0.5 seconds blank screen and finally either exactly the same image or a rendering (50% chance for each), for an unlimited period. At this point, observers were asked whether or not they saw or felt that there was a difference between the two consecutively shown pictures. If so, they were asked whether this difference seemed unnatural or not. No indication was given as to the kind of image differences to expect nor about the meaning of the word *unnatural*. Two levels of rendering were used: $\rho = 10\%$ in the first part of the experiment and $\rho = 15\%$ in the second part. We classified the observer's answers as follows: *true positive* (change detected between two different images), *true negative* (no change detected between two identical images), *false positive* (change detected between two identical images) and *false negative* (no change detected between two different images). Among the cases in which observers saw a difference, we counted the number of times that they found it unnatural. Figure 5 shows the results obtained.

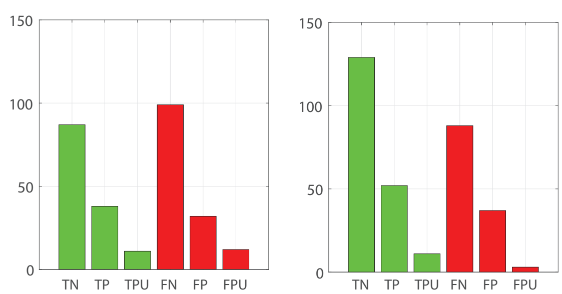

Figure 5. Results from the first subjective experiment (left: first sequence, right: second sequence): TN (*true negative*), TP (*true positive*), TPU (*true positive judged unnatural*), FN (*false negative*), FP (*false positive*) and FPU (*false positive judged unnatural*).

We used a two-sample binomial test [3] to evaluate whether the probability for a true negative is significantly different from that of a false negative (i.e., the probability that people can actually perceive a difference between original and rendered images). We found that, for the first image sequence ($\rho = 10\%$), there is no significant difference between the two probabilities (with a 95% confidence), whereas in the second image sequence ($\rho = 15\%$) there is. During the experiment, time was monitored. It took observers an average 5.4 seconds to answer the first question in the first sequence, and 3.9 seconds in the second sequence.

In addition to the fact there is a better chance for people to observe changes for a higher ρ, we assume that observers progressively moved from a naive to a task-driven viewing, thus invalidating the bottom-up saliency model used in our method as the experiment went. It also justifies the shorter answer time in the second session. Nevertheless, very few cases compelled observers to rate a change as unnatural, and we even recorded cases in which observers found the original image itself unnatural. Note also that we found that no scene took a significantly higher time for observers to judge than others.

B. Second experiment

A different group of 14 color-normal observers were asked to rate the difference between original and restored images. For each scene and for the same two ρ values as previously, both images were shown next to each other and observers were given three choices to rate the difference: *disturbing*, *acceptable* or *almost invisible*. Again, we let observers rely on their own interpretation of these words. Additionally, they were specifically instructed to let the operator know if they could not see any difference at all, although it never happened. Again, time was monitored and we measured that observers took an average 19.6 seconds to rate a pair of images in the first sequence ($\rho = 10\%$) and 11.6 seconds in the second sequence ($\rho = 15\%$). In total, observers found that the difference was *almost invisible* in 30 % of the cases, *acceptable* in 52% of the cases and *disturbing* in 18% of the cases only.

These results therefore indicate that the rendered images are, in a majority of the cases: 1) not noticeable at first glance, 2) not disturbing once noticed. As previously mentioned, the maximal amount of pixels that we were able to alter while remaining natural is about 15%.

VII. Conclusions and Future Work

This preliminary work demonstrated that change blindness can be exploited to help data reduction

Figure 4. Examples of renderings for $\rho = 15\%$. From left to right, column-wise: original, kept pixels (85%), restored images. Patch sizes - *Bird*: 15, *Horses*: 15, *Red raft*: 15, *Flowers (blue)*: 25, *Flower (yellow)*: 15, *Red leaf*: 25.

by introducing spatial redundancy without significantly disturbing the user. We defined *change blindness maps* and a compliant texture synthesis method to identify, remove and subsequently recover pixels subject to change blindness. Unlike comparable methods, ours is particularly intended for complex textures. We demonstrated via a subjective experiment that up to 15% of such an image can be altered in a way that is considered as natural and acceptable. More investigations should be carried out in order to determine the applicability of these findings for a wider variety of images, in which case other inpainting strategies should be tested as well.

References

[1] A. Borji and L. Itti. Exploiting local and global patch rarities for saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 478–485. IEEE, 2012.

[2] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207, 2013.

[3] L. Brown and X. Li. Confidence intervals for two sample binomial distribution. *Journal of Statistical Planning and Inference*, 130(1):359–375, 2005.

[4] A. Bugeau, M. Bertalmío, V. Caselles, and G. Sapiro. A comprehensive framework for image inpainting. *Image Processing, IEEE Transactions on*, 19(10):2634–2645, 2010.

[5] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[6] C. Guillemot and O. Le Meur. Image inpainting: Overview and recent advances. *Signal Processing Magazine, IEEE*, 31(1):127–144, 2014.

[7] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (TOG)*, 26(3):4, 2007.

[8] K. He and J. Sun. Statistics of patch offsets for image completion. In *Computer Vision–ECCV 2012*, pages 16–29. Springer, 2012.

[9] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.

[10] J.-S. Lee and T. Ebrahimi. Perceptual video compression: A survey. *Selected Topics in Signal Processing, IEEE Journal of*, 6(6):684–697, 2012.

[11] J. Li, M. D. Levine, X. An, X. Xu, and H. He. Visual saliency based on scale-space analysis in the frequency domain. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):996–1010, 2013.

[12] I. Lissner and P. Urban. Toward a unified color space for perception-based image processing. *Image Processing, IEEE Transactions on*, 21(3):1153–1168, 2012.

[13] D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang. Image compression with edge-based inpainting. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(10):1273–1287, 2007.

[14] S. R. Mitroff, D. J. Simons, and D. T. Levin. Nothing compares 2 views: Change blindness can occur despite preserved access to the changed information. *Perception & Psychophysics*, 66(8):1268–1281, 2004.

[15] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *Image Processing, IEEE Transactions on*, 21(12):4695–4708, 2012.

[16] A. I. Oncu, F. Deger, and J. Y. Hardeberg. Evaluation of digital inpainting quality in the context of artwork restoration. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 561–570. Springer, 2012.

[17] J. K. O'Regan and A. Noë. A sensorimotor account of vision and visual consciousness. *Behavioral and brain sciences*, 24(05):939–973, 2001.

[18] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.

[19] F. Racapé, O. Déforges, M. Babel, and D. Thoreau. Spatiotemporal texture synthesis and region-based motion compensation for video compression. *Signal Processing: Image Communication*, 28(9):993–1005, 2013.

[20] S. D. Rane, G. Sapiro, and M. Bertalmio. Structure and texture filling-in of missing image blocks in wireless transmission and compression applications. *Image Processing, IEEE Transactions on*, 12(3):296–303, 2003.

[21] D. J. Simons and M. S. Ambinder. Change blindness theory and consequences. *Current directions in psychological science*, 14(1):44–48, 2005.

[22] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *Signal Processing Magazine, IEEE*, 18(5):36–58, 2001.

[23] C.-W. Tang, C.-H. Chen, Y.-H. Yu, and C.-J. Tsai. Visual sensitivity guided bit allocation for video coding. *Multimedia, IEEE Transactions on*, 8(1):11–18, 2006.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error measurement to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004.

[25] Z. Xiong, X. Sun, and F. Wu. Block-based image compression with parameter-assistant inpainting. *Image Processing, IEEE Transactions on*, 19(6):1651–1657, 2010.

[26] J. Zhang and S. Sclaroff. Saliency detection: A boolean map approach. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 153–160. IEEE, 2013.

[27] L. Zhang, Y. Shen, and H. Li. VSI: A visual saliency induced index for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 23(10):4270–4281, 2014.