Vision, Modeling, and Visualization (2021)
B. Andres, M. Campen, and M. Sedlmair (Eds.)

Context Aware Exemplar-based Image Inpainting using Irregular Patches

C. Fotsing[1] and D. Cunningham[1]

[1]Brandenburgische Technische Universität Cottbus-Senftenberg,Chair of Graphic Systems, Germany

Abstract

We propose a new exemplar-based image inpainting method in this paper. Our method is based on the Criminisi pipeline. We focused on three main stages of the pipeline; calculation of priorities, construction of patches, and the search for the best match. To assign a high priority to patches constructed from the edge pixels, we use the ability of segmentation algorithms to divide an image into different texture blocks. The patches built from pixels located at the border between several texture blocks receive a high priority. Unlike most patch-based image inpainting methods which use regular patches (rectangle, square), the shape and size of our patches depend on the textural composition around the original pixel. The patches are built using a region growing principle in the different texture blocs around the original pixel. The search for the best match is done contextually. We search for the best match of the patch with the highest priority in a similar environment to its neighborhood around the target zone. The method is simple and easy to implement. The experiments show that our method obtains more plausible results than the basic method of Criminisi and its improved version Amoeba in most cases.

Keywords: Inpainting, Exemplar-based, Color segmentation, Region growing

1. Introduction

Digital image inpainting is an important task in computer vision that aims to restore damaged or missing parts of an image. It is used in several applications ranging from the restoration of artworks (paintings, films, photographs) [BBS01, KMFR95] to the removal of unwanted artifacts from images and videos (object removal) [CPT04a, LMWY13]. Unlike other image restoration problems like denoising where the pixels to be processed contain both the correct information and the bias, in image inpainting, we do not have any information on the value of pixels to be inpainted. The inpainting problem is described as follows: Let I be an Image of size $m \times n$. Mathematically, I can be defined as a two-dimensional function

$$\begin{array}{rcccc} I & : & M \times N & \rightarrow & R^k \\ & & (x,y) & \mapsto & I(x,y) \end{array}$$

where k is the number of color channels (1 for gray images and 3 for RGB images). $M = \{1,2,...,m\}$ and $N = \{1,2,...,n\}$. Within the framework of the inpainting problem, $\exists \Omega \subset M \times N$, $\forall (x,y) \in \Omega \quad I(x,y) = ?$. In the literature, Ω is called the target zone, $M \times N \setminus \Omega$ the source zone denoted by Φ, and the set of border pixels between the two regions is denoted by $\delta\Omega$. The objective of inpainting is to reconstruct the unknown pixels in a way that is not detectable to observers, i.e., the result should appear natural to the human eye and should be as physically plausible as possible [CPT04a].

To solve inpainting problems, researchers proposed a wide range of solutions based on different approaches [GL14]. One of the simplest and most effective solutions is the exemplar-based algorithm proposed in [CPT04a]. It is a simple-well-defined pipeline that gradually and iteratively fills the vacuum area Ω. Once Ω and Φ are defined, the algorithm performs 4-main steps at each iteration: 1- it calculates the priorities of the candidate patches, 2- it searches for the best match for the candidate patch with the highest priority, 3- it propagates structures and textures into Ω, and finally 4- it updates Ω, Φ, and $\delta\Omega$. The method's effectiveness primarily depends on the reliability and precision of the processes used at each stage of the pipeline. Our main objectives in this work are: (1) to propose a new image inpainting method based on this pipeline. (2) to compare the results of our method with those of other methods.

The research has three main contributions. Firstly, we propose a new formula for calculating the priority of candidate patches. We use the reliability of image segmentation algorithms (ability to determine the borders between the different texture blocks of an image) to assign high priorities to candidate patches constructed from the texture border pixels (edge pixels). This ensures the continuity of the image structures in Ω. Secondly, instead of using geometrically regular patches (squares, rectangles) like in traditional exemplar-based methods, we execute a local region growing process to build up the patches efficiently. This approach reduces the likelihood of copying inconsistent pixels. Finally, once the candi-

DOI: 10.2312/vmv.20211373

date patch with the highest priority is determined, the search for the best match in Φ is performed contextually, i.e., we are not only looking for the best match of the patch, but the best match in a context similar to the one surrounding the patch in the target area. This contextual approach improves the consistency between Φ and the new texture created to fill Ω.

The paper is organized as follows. In section 2, we review some relevant works on image inpainting. We present the new method in section 3. Section 4 describes the experimental procedures and analyzes the results. The conclusions of the research are presented in section 5.

2. Related works

Restoration of paintings is an activity as old as art. It consists of restoring damaged paintings "by hand" (loss of paint, weakened canvas, tears, water damage, or fire) so that the result is as similar as possible to the original work or so that the changes are imperceptible and consistent [CRG*13]. It is a tedious task and requires an artistic background and a lot of concentration. Researchers have recently proposed several digital solutions that attempt to virtually replicate the basic techniques used by professionals to recover the damaged parts of images [PPM12]. These solutions can be classified into two algorithmic classes: sequential algorithms [GL14] and deep learning [SWFY20, PKD*16].

2.1. Sequential methods

Depending on the used approach, sequential methods can be grouped into three sub-classes: diffusion-based methods, exemplar-based methods, and methods using both approaches.

2.1.1. Diffusion-based inpainting

Diffusion-based methods use principles similar to physical propagation phenomena to locally propagate information from Φ into Ω. The basic principle consists in exploiting information from the neighborhoods of the border pixels $\delta\Omega$ to smoothly extend Φ into Ω while taking care to preserve the orientation of the isophote lines [GL14]. Various mathematical models have been used in the framework of diffusion-based image inpainting. They include Partial Differential Equations (PDE) [Sch15], Fourier Transform [MT19], wavelets [CDNL98, DJL*12], Cahn-Hilliard Equation [BHS09, BEG07], statistical and stochastic modeling [GG84, LZW03].

In recent decades, the diffusion-based methods that have received the most attention are those based on partial differential equations. The pioneering work in the field is the method proposed by Bertalmío et al. [BSCB00]. After identification of the Ω area by the user, they used an anisotropic diffusion model to smoothly propagate neighborhood information from the border pixels $\delta\Omega$ into Ω. To ensure the continuity of the isophote lines, the propagation was done according to the normal of the gradient vector of the pixels along the border $\delta\Omega$. The main drawbacks of this technique are its slowness and the difficulty to restore large textured regions. In order to improve the computational time, Telea [Tel04] presented a Fast Marching Method for image inpainting. They diffused an image smoothness estimator along the image gradient, similar to [BBS01]. The image smoothness was estimated as a weighted average over a known image neighborhood of the pixel to be inpainted. To maintain the isophote lines direction, other PDE-based models have been suggested, including Total Variational (TV) [SC02] and Curvature-Driven Diffusion [CS01].

Diffusion-based methods are suitable for completing straight lines, curves and for inpainting small areas and, they avoid having unconnected edges. However, they are not well suited to recover the texture of large surfaces [GL14].

2.1.2. Exemplar-based inpainting

The general principle behind this approach is based on the idea that it is possible to consistently restore damaged parts of an image by filling unknown pixels with color values from the source area. Pixels can be restored in blocks (patch-based inpainting) [BDTL15] or individually (pixel-based inpainting) [WL00a, DSC04].

Giving priority to edge pixels, Qiang et al. [QHX17] restored Ω pixel by pixel. After selecting border pixels with the highest priority at each iteration, they constructed (searched for) a subset of candidate patches for each selected pixel P_i similar to the patch centered at P_i. The value of the center pixel of each candidate patch is a possible value of P_i. The median method is adopted to select the best filling value of P_i. As in other pixel-based methods, this solution suffers from high computational costs and has difficulty restoring large textures made up of many small objects [GL14].

To speed up the restoration process, an obvious solution is to proceed by copying sub-regions of pixels (patches) at the same time. One of the most promising patch-based methods was proposed by Criminisi et al. [CPT04a]. As discussed above, it is a pipeline made up of four main steps that restore Ω by gradually sampling and copying sub-regions of color values from the source region Φ. To ensure the propagation of the image structures in Ω, they assigned a high fill order priority to patches containing edges. The hindrance of this method is the propagation of synthetic errors. i.e., copying a few unreliable pixels in one step is enough to make the result inconsistent and implausible. Over the past two decades, many improvements have been proposed. In most solutions, the general idea of the pipeline remains unchanged. Only the approaches used at one or more stages of the pipeline are improved or modified [FZ18, OLKK19].

To overcome the problem of discontinuous structures and inconsistent textures, several authors proposed improvements. Xu and Sun [XS10] designed a patch structure sparsity function to assign high priorities to patches located at the image structure. Instead of directly copying the patch with the best match, they built the patch to be filled as the sparse linear combination of candidate patches under the local patch consistency constraint in a framework of sparse representation. Lu et al. [LHLC10] suggested using adaptive patch sizes according to structure and the local texture. After calculating the best match, Castillo et al. [CCWB18] used a region growing process (amoeba) to extract only consistent and reliable pixels instead of copying all the pixels in the patch.

2.1.3. Hybrid methods

Hybrid methods aim to combine the two previous approaches by simultaneously taking advantage of the capacity of the diffusion-based methods to preserve the structure of images and the exemplar-based methods to reconstruct large textures. The idea behind is to separate images into several components (e.g., structure, texture), then to restore them individually using the most suitable approach and combine the results, or combine the different approaches into a global function.

Bertalmio et al. [BVSO03] suggested to decompose images into the sum of two functions (structure and textures), then to reconstruct each of the components separately using a suitable filling-in algorithm, and finally combine the two output components to have the result. Following the same pipeline, Grossauer [Gro04] proposed to fill-in the structure component with the PDE-based solution of [GS03]. He employed the texture synthesis algorithm in [WL00b] to restore the texture component. Wu and Ruan [JQ08] used a bi-directional diffusion PDE to inpaint the structure after separating the structure part from the texture part with the total variation equation. The texture was restored using an exemplar-based inpainting solution constrained by a cross-isophote diffused data term.

Unlike the previous solutions, which separated the decomposition and filling-in stages, Elad et al. [ESQD05] combined the two stages in a single task. The separation was done using the morphological component analysis (MCA) algorithm proposed in their previous work to decompose the image into texture and cartoon layers. They modeled the inpainting problem as an optimization. Bugeau et al. [BBCS10] proposed to combine the texture synthesis term, the diffusion term, and a third term (coherence term) into an energy function. The restoration was executed by minimizing the energy function.

2.2. Deep learning techniques

With the recent evolution in deep learning, several learning models have been proposed to restore damaged parts of digital images. The deep learning methods used for image inpainting are mainly based on traditional Convolutional Neural Networks (CNN) or Generative Adversarial Networks (GAN).

[KSSH14, CSL*17, KY15]proposed to first train a neural network model to automatically map the unknown pixels without user intervention (blind inpainting). The model suggested by Cai et al. [CSL*17] is a CNN with three convolutional layers, which takes as input a damaged image, identified the corrupted or unknown pixels, and automatically restores them. The model was trained using Stochastic gradient descent with standard backpropagation. After extracting the missing regions, Alilou and Yaghmaee [KY15] sorted them according to their size and then applied a pre-trained Generalized Regression Neural Network (GRNN) model for restoration. To restore shape with sharp structures and fine-detailed textures Yan et al. [YLL*18] incorporated a special shift-connection layer with guidance loss to the U-Net architecture. They took into account the shape of the missing region as an important parameter in the recovering process.

Introduce by Goodfellow et al. [GPAM*14] to perform generative modeling, GANs have been used by several authors to solve image inpainting problems [YLY*18, VSB19]. Pathak et al. [PKD*16] proposed an unsupervised visual feature learning algorithm driven by context-based pixel prediction to preserve the appearance and the semantics of visual structures in images. They trained a CNN model to predict missing pixels based on their surrounding context. Their model is a simple encoder-decoder pipeline. The role of the encoder is to extract a latent feature representation of the image and then using these features the decoder restores the unknown regions. To make the prediction look real, they used an adversarial loss function similar to [GPAM*14]. Focusing on both the local and global consistency of the inpainted image, Iizuka et al. [ISSI17] presented a learning model for image completion. Their architecture consists of a completion network and two auxiliary context discriminator networks used only for training the completion network. During the training process, the discriminators' role is to check if the inpainted image generated by the completion network is real, while the completion network is trained to fool both discriminator networks.

3. Method

We present a new image inpainting method based essentially on the [CPT04b] pipeline in this section. The method focuses on three main aspects: the calculation of the priorities, the construction of the patches, and the search for the best match. The main idea is to use the abilities of image segmentation algorithms (boundary determination between different texture blocks of an image) to assign high priority to patches constructed from border pixels located on the edges. Then, we use a region growing principle to build the pixel subset (patch), which has to be extended in Ω. Instead of searching only for the best patch match in Φ, we search for the best match in an area similar to the initial neighborhood of the patch around the target zone.

3.1. Priorities calculation

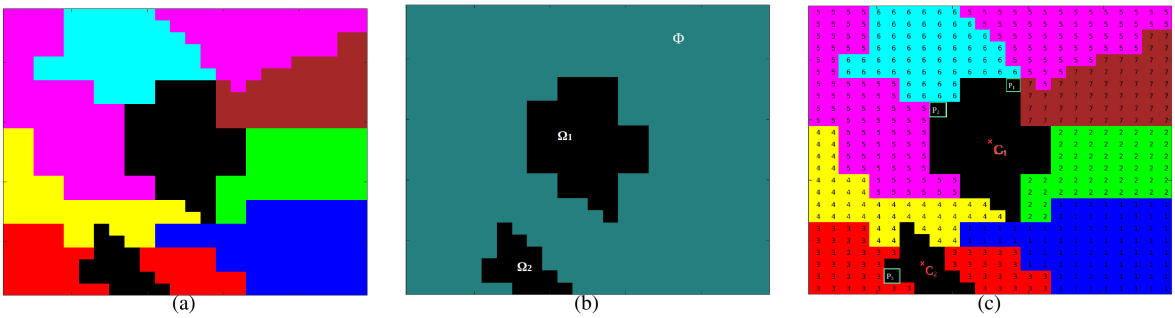
To ensure the continuity of image structures in Ω, it is necessary to give a high priority of fill order to the patch containing edges [CPT04b]. Based on this observation, we proposed a new formula to calculate priorities. Let $p_i(x_p, y_p)$ be a pixel belonging to the border $\delta\Omega$, the priority of the patch constructed from p_i is

$$P(p_i) = N(p_i) \times Distance(p_i, Center) \times Rate(p_i)/Conf(p_i) \quad (1)$$

where:

$N(p_i)$ corresponds to the number of texture blocks surrounding the pixel p_i. It is determined using the color image segmentation process [GGGD14]. As shown in Figure 1(c) $N(p_1) = 3$, $N(p_2) = 2$, $N(p_3) = 1$. This implies that around pixels p_1, p_2, and p_3 there are 3, 2, and 1 texture blocks, respectively. $N(p_i)$ assigns a high priority to the patches built from border pixels located at the edges. This ensures the propagation of image structures in Ω.

$Distance(P_i, Center)$ is the Euclidean distance between P_i and $Center$. In our approach, we have divided Ω into a disjoint subsets $\Omega_1, \Omega_2, ..., \Omega_l$ as illustrated in Fig 1(b). Let $C_i(X,Y)$ be the center of the subset $\Omega_i = \{p_1(x_1,y_1), p_2(x2,y2), ..., p_n(x_n,y_n)\}$: $X = \frac{1}{n}\sum_{k=1}^{n} x_i$, $Y = \frac{1}{n}\sum_{k=1}^{n} y_i$. Assume that P_i is at the border of Ω_i. Then, $Distance(P_i, C_i)$ is the Euclidean distance between P_i and



(a) (b) (c)

Figure 1: *Illustration of a segmentation case. a)* 24×24 *generated image, b) mask, c) segmented image*

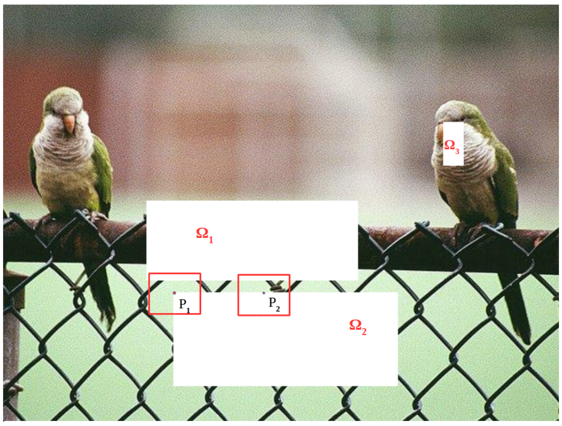


Figure 2: *Known pixels priority box:* $rate(P_1) > rate(P_2)$ *(see Equation 1)*

C_i. This parameter promotes filling (restoration) from the outside to the inside of Ω.

$Rate(P_i)$ is the rate of known pixels in a predefined maximum radius. This parameter gives filling priority to areas with a high known pixels ratio. It is calculated by dividing the number of known pixels contained in the delimited perimeter by the total number of pixels in the space. As shown in Figure 2, the box built around P_1 contains more known pixels than the one built around P_2, so $Rate(P_1) > Rate(P_2)$.

$Conf(P_i)$ prevents reconstructions that suddenly prolong the edges in Ω. For example, in Figure 2, a rapid extension of the grid inside Ω without considering the green texture would result in an inconsistent restoration. In the initial state, each border pixel has confidence of 1. Let K be the subset of border pixels created after filling a patch constructed from a pixel P with $Conf(P) = i$. $\forall p_i \in K, Conf(p_i) = i+1$.

3.2. Patch construction

Once the pixel with the highest priority has been determined, we construct the subset of the known pixels that must be extended into Ω. Unlike traditional methods that propose building a regular patch (circle, square, rectangle) centered around the border pixel, we suggest constructing an irregular patch using a region-growing technique. Let P be the border pixel with the highest priority. The principle is to grow a region in each texture block around P and finally merge the resulting blocks into one set (see Figure 3). The growth process in a block starts from a pixel C, a neighboring pixel of P belonging to the corresponding block. The basic principle of the growing process is inspired by the morphological amoeba algorithm [LDM05]. Algorithm 1 describes the main idea behind the process. It takes as input an image to restore *GrayIm*, the central pixel from which the growth is made P_k, a cumulative difference threshold $\beta > 0$ which is used to manage the luminosity variation in the texture block, a rate λ used to control the patch extent according to the image size, and the *Mask*. We start by marking all the pixels of the image as unprocessed. P_k is marked as belonging to the growing region *SegBlock*. Each of the neighboring pixels to P_k (we denote them P_j) belonging to Φ receives as cumulative difference $P_j.cum\_diff = |GrayIm(P_k) - GrayIm(P_j)|$. In Algorithm 1, these steps are computed by the functions $ComputeCum\_diffNeighbour()$. Then, all the P_j are stacked in a stack *Stk* and marked as treated. As long as *Stk* is not empty, we unstack a Pixel P_i. If $P_i.cum\_diff < \beta$ and $|P_i.x - P.x| <= \lambda * GrayIm.Width$ and $|P_i.y - P.y| <= \lambda * GrayIm.height$, P_i is put into *SegBlock*. All the untreated neighboring pixels of P_i belonging to Φ (we note them P_l) receive a value $P_l.cum\_diff = P_i.cum\_diff + |GrayIm(P_i) - GrayIm(P_l)|$. Each P_i is then marked as processed and stacked in *Stk*. Once the process is applied to all the texture blocks in the direct neighborhood of P, the final patch is the union of the resulting *SegBlock*s.

3.3. Best match determination

The search for the best match is performed contextually. We find the best match in an environment similar to the home region (a neighborhood similar to the one surrounding the patch around Ω). This avoids copying regions surrounded by neighborhoods different from the original one. We define a context with a predefined dimension around the template (the match). Note that the context size may be reduced depending on the location of the template. For example, templates located at the borders of the image frame can only be surrounded by a smaller context. As shown in Figure 4, we see that the best match which does not take into account the en-

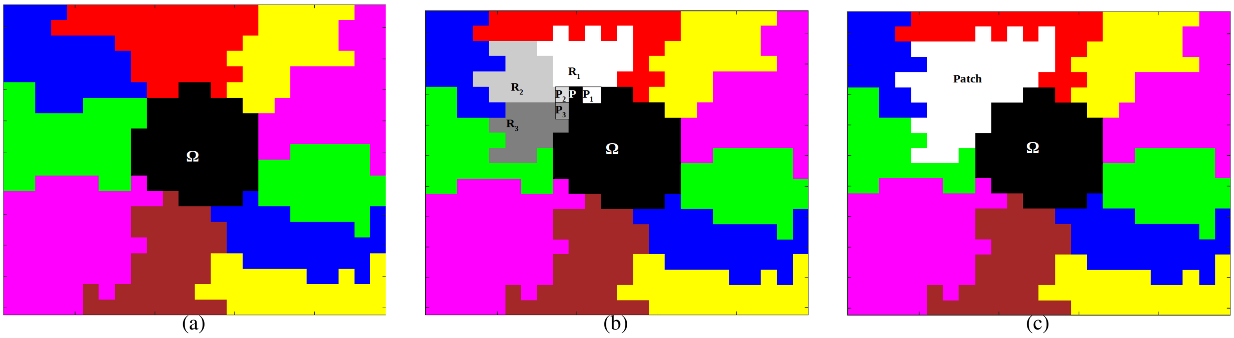


Figure 3: *Patch growing. a) 24×24 generated image with target area Ω, b) growing regions:P is the border pixel with the highest priority, R_1,R_2 and R_3 have been grown from P_1, P_2 and P_3 respectively, c) Patch $= R_1 \bigcup R_2 \bigcup R_3$*

Algorithm 1 *Region growing: GrayIm, Mask,P,P_k, $\beta > 0, \lambda > 0$*

$Stk \leftarrow \emptyset$
$Segblock \leftarrow \{P_k\}$
$W = GrayIm.width$
$H = GrayIm.height$
$N \leftarrow ComputeCum\_diffNeighbour(P_k, GrayIm, Mask)$
$MarkAsTreated(N)$
$Stack(Stk, N)$
while $Stk \neq \emptyset$ **do**
  $P_i \leftarrow Unstack(Stk)$
  if $P_i.cum\_diff < \beta$ && $|P_i.x - P.x| < \lambda \times W$ && $|P_i.y - P.y| < \lambda \times H$ **then**
    $Segblocks \leftarrow Segblock \bigcup \{P_i\}$
    $N \leftarrow ComputeCum\_diffNeighbour(P_i, GrayIm, Mask)$
    $MarkAsTreated(N)$
    $Stack(Stk, N)$
  end if
end while
return *Segblock*

---

vironment ($Match_1$), would lead to an incoherent extension of the green texture block into the gray metal bar.

Another context with the same dimension is defined around the patch in its original environment. As shown in Figure 5, the area delimited by Γ contains both the patch P and the neighborhood V. $\Gamma = P \cup V$, $P \cap V \cap \Omega = \emptyset$ and $||\Gamma|| = ||\Gamma'||$. The best match is the one that minimizes $SSD(P,P') * SSD(V,V')/|V|$. With $V \neq \emptyset$ and $SSD()$ refering to the sum of squared differences.

3.4. Patch extension and update

Once the best match is found, we perform region growing from the correspondents of the different C_i pixels in the best match area (see Figure 6(b)). The union of the different resulting growing regions corresponds to an extension of the patch. Then, Ω is filled with the subset of the excess pixels (see Figure 6(c)). Ω, Φ and $\delta\Omega$ are then updated. Let P be the current border pixel from which the patch is built and, $R = \{P_1, P_2, ..., P_n\}$ the set of the border pixel generated after partially filling Ω. $\forall P_i \in R, Conf(P_i) = Conf(P) +$

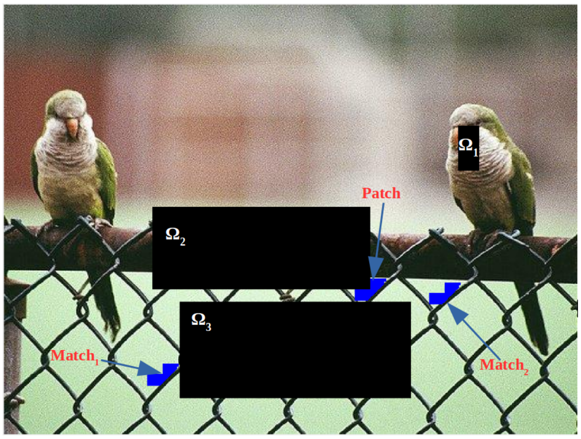


Figure 4: *Illustration of contextual and non-contextual matching: $Match_2$ is obtained using the context and $Match_1$ without the context.*

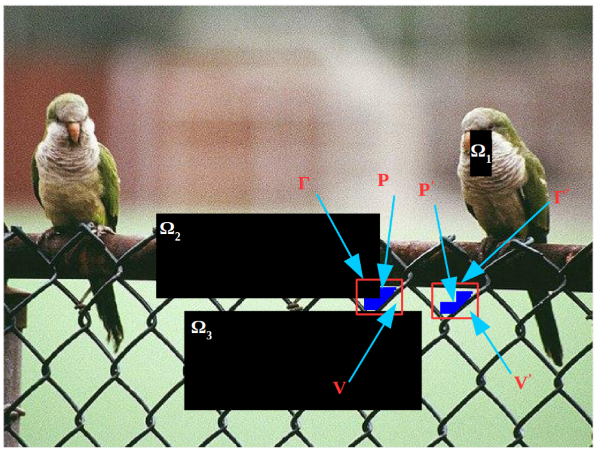
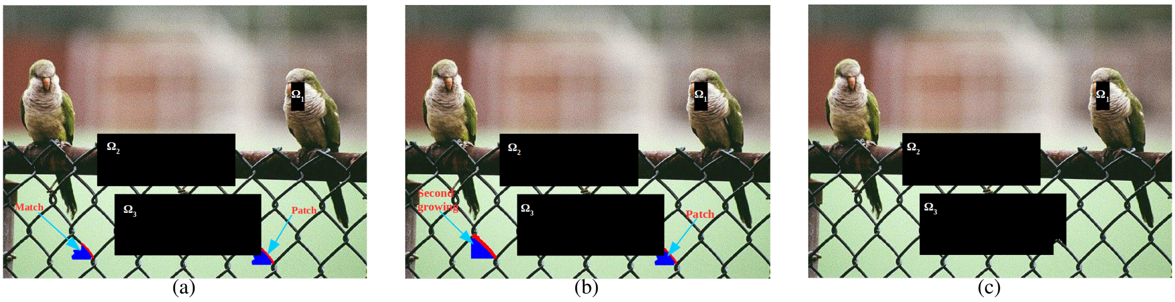
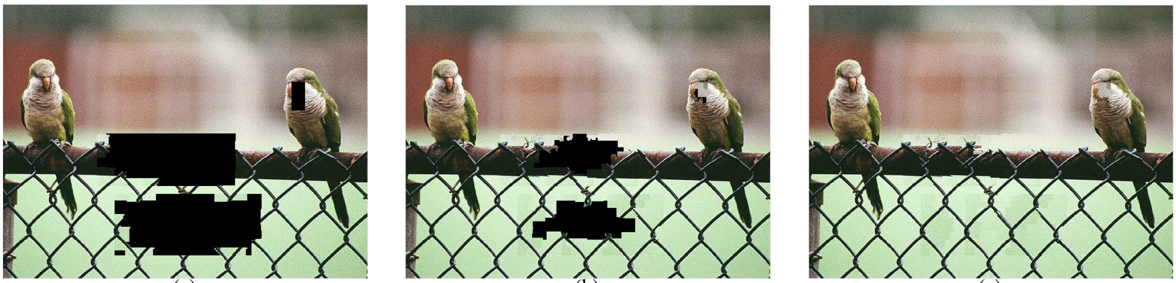


Figure 5: *Best match with context: $\Gamma = P \cup V$, $\Gamma' = P' \cup V'$, $P \cap V \cap \Omega = \emptyset$, $P' \cap V' \cap \Omega = \emptyset$, and $||\Gamma|| = ||\Gamma'||$.*



(a) (b) (c)

Figure 6: *Patch extension and update. a) Best match, b) Second region growing, c) Update image*



(a) (b) (c)

Figure 7: *Some steps of the restoration process*

1. The process is repeated until we have no more border pixels. Figure 7 presents the progress of the restoration at some stages.

4. Experiments

To demonstrate the effectiveness of our method, we carried out several experiments. The datasets used in our experiments are images from [CCWB18]. This benchmark is known for containing images with attributes that present challenges in preventing artifact generation and preserving content and structure.

Instead of defining patches using a regular shape as in most Criminisi-based inpainting methods, we controlled the patch extent using two parameters λ and the maximal cumulative difference β (see algorithm 1). We submitted 16 incomplete images to restore (see Fig 8) to three exemplar-based inpainting methods; the original method of Criminisi et al. [CPT04b], the modified version (amoeba) proposed by Castillo et al. [CCWB18], and to our method. For the Criminisi algorithm, 19 restorations were performed on each image using different patch sizes ranging from 2 to 20. The restorations with the amoeba method were performed using the maximum amoeba distance TH set to 20 and a physical distance $PD = 1$. We tested different radii ranging from 1 to 10 for each image. For our method, the maximal cumulative distance β was set to 0.15 in the *LAB* color space. For each image, we carried out 20 reconstructions with a random value of λ selected between 0.01 and 0.1. The context was a $3*\lambda*W \times 3*\lambda*H$ rectangle centered on the border pixel with the highest priority (W and H are the width and the height of the image, respectively). Fig 8 shows images of the benchmark.

4.1. Results and validation

The main goal of inpainting algorithms is to improve image filling quality so that any imperfection is not noticeable by a person who is not familiar with the original image. The ideal is to let the results be appreciated by human observers in order to judge the performance of the algorithms. To get around this tedious task, we used three metrics: Peak Signal-to-Noise Ratio (PSNR) [MM98] , Edge Histogram (EH) [WPP02] and Structural Similarity Index(SSIM) [WB02].

Peak Signal-to-Noise Ratio (PSNR) is a measure of distortion used in image processing to quantify the performance of image restoration or compression algorithms. It is an estimate of the restored image quality compared with the original image. Let A be a restored image and B the corresponding original image, both with size $M \times N$.

$$PSNR(A,B) = 10\log\left(\frac{S^2}{MSE(A,B)}\right) \tag{2}$$

where $S = 255$ for an 8-bit image and

$$MSE(A,B) = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}[A(i,j)-B(i,j)]^2 \tag{3}$$

Edge Histogram: The EH descriptor represents the distribution of 5 types of edges (vertical, horizontal, 45-degree diagonal, 135-degree diagonal, and non-directional edges) in each local area called a sub-image. The image is divided into a grid of 4×4 blocks. Whatever the size of the image, the final descriptor is an edge histogram with 150 bins (80 bins (local) + 5 bins (global) + 65 bins

Figure 8: *Benchmark: In pink the target areas*

(semi-global)) [WPP02]. Let A and B be two images, the distance $D(A,B)$ between their edge histogram can be measured by

$$D(A,B) = \sum_{i=0}^{79} |(Local\_A[i] - Local\_B[i])| + 5 \times \sum_{i=0}^{4} |(Global\_A[i] - Global\_B[i])| \sum_{i=0}^{64} |(Semi\_Global\_A[i] - Semi\_Global\_B[i])| \quad (4)$$

Structural Similarity Index(SSIM) is a human visual system (HVS) based metrics introduced by Wang and Bovik [WB02] to assess the human visibility similarity between a restored image and the original. SSIM measures the similarity of the combination of contrast and luminance [ANC12]. Let A and B be the original image and the restored one. Both images are first divided into blocks of size 8×8 and converted into vectors. Let $x = \{x_1, x_2, ..., x_T\}$ and $y = \{y_1, y_2, ..., y_T\}$ be two corresponding vectors from A and B respectively:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_x^2 + C_2)} \quad (5)$$

where C_1 and C_2 are constants. μ_x, μ_y are the mean values of the vectors x and y. σ_x^2 and σ_y^2 the variances and, σ_{xy} the covariance between x and y.

Let L be the number of local windows over the images. So $A = \{A_1, A_2, ..., A_L\}$ and $B = \{B_1, B2, ..., B_L\}$ then,

$$SSIM(A,B) = \frac{1}{L} \sum_{i=1}^{M} SSIM(A_i, B_i) \quad (6)$$

Table 1 presents the values of parameters that produce the best result for each image of the benchmark for each of the three metrics. Since EH reflects the distance between the original image and the inpainted one, the optimal restoration is the result that generates

Table 1: *Best metric values obtained by the methods are presented in bold font: The parameters that produce the best results are indicated in the brackets (best radii for Criminisi and Amoeba and, $\lambda * 10^3$ for our method). For the EH metric, the smaller values indicate better reconstructions while for PSNR and SSIM larger values indicate better reconstructions.*

		Criminisi			Amoeba			Our method		
Image	Size ($W \times H$)	PSNR	SSIM	EH	PSNR	SSIM	EH	PSNR	SSIM	EH
Bicycle	460×300	**24.13**(10)	**0.84**(3)	**18.71**(9)	**24.5**(5)	**0.95**(4)	**14.62**(11)	**24.69**(15)	**0.96**(30)	**15.61**(25)
Twobirds	600×450	**16.45**(5)	**0.71**(1)	**39.77**(16)	**17.25**(12)	**0.75**(1)	**27.27**(12)	**22.41**(25)	**0.94**(25)	**25.78**(25)
BattleShip	1024×756	**22.16**(11)	**0.85**(6)	**46.01**(13)	**23.02**(8)	**0.87**(4)	**44.20**(8)	**23.4**(45)	**0.84**(15)	**47.91**(45)
Blueman	1024×681	**18.5**(7)	**0.91**(2)	**17.53**(9)	**18.67**(11)	**0.94**(6)	**18.25**(16)	**19.02**(30)	**0.95**(30)	**17.98**(15)
BrickHouse	1024×683	**24.43**(15)	**0.89**(5)	**31.86**(13)	**24.28**(6)	**0.84**(8)	**38.55**(6)	**24.50**(35)	**0.91**(35)	**31.82**(35)
Bungee	206×308	**17.06**(4)	**0.80**(8)	**38.41**(5)	**16.78**(19)	**0.81**(20)	**35.06**(19)	**17.08**(25)	**0.82**(25)	**29.78**(25)
Castle	1024×768	**23.87**(9)	**0.91**(11)	**33.72**(9)	**23.89**(13)	**0.94**(9)	**39.24**(2)	**23.84**(40)	**0.97**(35)	**32.80**(60)
Cat	1024×683	**23.2**(18)	**0.97**(9)	**42.74**(15)	**23.64**(3)	**0.98**(1)	**35.48**(14)	**23.59**(10)	**0.98**(25)	**38.75**(25)
Child	1024×680	**25.63**(5)	**0.95**(13)	**30.85**(16)	**25.73**(13)	**0.97**(1)	**28.29**(2)	**25.64**(25)	**0.96**(55)	**29.68**(20)
Eagle	600×402	**28.57**(20)	**0.94**(2)	**27.26**(17)	**28.46**(7)	**0.95**(1)	**19.91**(20)	**29.36**(15)	**0.95**(10)	**29.78**(55)
fish	1024×768	**24.14**(5)	**0.84**(15)	**30.51**(16)	**24.16**(2)	**0.91**(6)	**28.29**(2)	**24.43**(30)	**0.91**(10)	**30.66**(25)
matrioska	1440×971	**19.56**(13)	**0.74**(11)	**27.69**(16)	**19.58**(17)	**0.89**(1)	**25.84**(2)	**19.34**(45)	**0.85**(15)	**29.43**(30)
mochizuki	547×346	**24.04**(18)	**0.89**(1)	**19.89**(19)	**22.47**(9)	**0.82**(1)	**16.71**(17)	**25.27**(10)	**0.94**(50)	**18.80**(25)
mountains	512×683	**27.53**(9)	**0.94**(17)	**8.32**(15)	**28.81**(17)	**0.97**(15)	**10.43**(4)	**28.51**(15)	**0.97**(20)	**9.80**(15)
penguins	615×461	**19.71**(13)	**0.86**(15)	**18.96**(19)	**19.31**(19)	**0.91**(8)	**18.8**(11)	**19.61**(10)	**0.92**(35)	**17.71**(15)
car	500×375	**24.08**(17)	**0.91**(3)	**8.34**(3)	**24.19**(8)	**0.95**(2)	**10.44**(3)	**25.06**(15)	**0.96**(10)	**8.32**(15)

Table 2: *Average values of metrics over all of our experiments*

	Criminisi	Amoeba	Our method
PSNR	22.69	22.79	**23.48**
SSIM	0.87	0.90	**0.92**
EH	27.53	**25.71**	25.91

the lowest value. However, for PSNR and SSIM the best results are those that generate the highest values.

Criminisi results: The results in table 1 show that the patch radius of the best reconstruction is different for each metric in all cases. For the SSIM metric, the patches that generate the best values are generally below 10, which is the reverse for the EH metric where the best values of the patch sizes are mainly between 13 and 19. In the case of the PSNR metric, the patches producing the best values do not have a specific interval. In general, it is difficult to predict the radius for each metric that yields the value.

Amoeba results: As in Criminisi, the metrics do not unanimously indicate the best reconstruction for all cases. The PSNR and EH metrics tend to produce the best values for large patches (> 10), while SSIM generates the best values for small patches (in most cases close to 1).

Our results: Except for the Twobirds image where the metrics agree on the λ value generating the best reconstruction, the parameter values are divergent for the other images. However, there are several cases where at least two of the metrics indicate the same value as the one having the best reconstruction. In general, the best values of λ oscillate between 0.01 and 0.03 for each of the metrics.

An analysis of the results of each image reveals that our algorithm produces the best restoration in 13 of the 16 cases for the PSNR metric, see Table 1. For the SSIM metric, our method is the best in 14 cases and in 6 cases for the EH metric. The average value obtained for each of the metrics on the benchmark is presented in Table 2. We note that our method wins over the two others according to the PSNR and SSIM metrics. However, the amoeba method is slightly better for the EH metric. In general, these results reflect an improvement in the restoration quality obtained with our method compared to those of Amoeba and Criminisi.

The best results (subjective judgment) generated by each of the three methods for some images are presented in Figure 9. These results do not correspond to those indicated by the metrics in all cases. We can see that the results obtained by our method are the most plausible in the majority of cases.

4.2. Conclusion

A new exemplar-based inpainting method was presented in this work. Based on Criminisi's pipeline, our method proposes several improvements at each stage of the restoration process. A new priority function was defined. To give a high order priority to the patches constructed from the pixels located on the edges, we used the ability of color segmentation algorithms to subdivide images to determine the patches covering the largest number of texture blocks. This approach allowed for a considerable improvement in the reconstruction of continuous structures in the image. Unlike in most exemplar-based inpainting methods where patches are usually defined by regular structures (square, rectangle, ...), in the new method, the patches are built using a region-growing algorithm in the different texture blocks surrounding the border pixel with the highest priority. The search for the best match is done contextually. We look for the best match in a neighborhood similar to the one surrounding the patch around the target area.

To demonstrate the effectiveness of our method we used a benchmark of 16 images. A comparison of our results with those obtained by the Criminisi and Amoeba algorithms shows a clear improvement in the quality of the restoration obtained by our method. Our

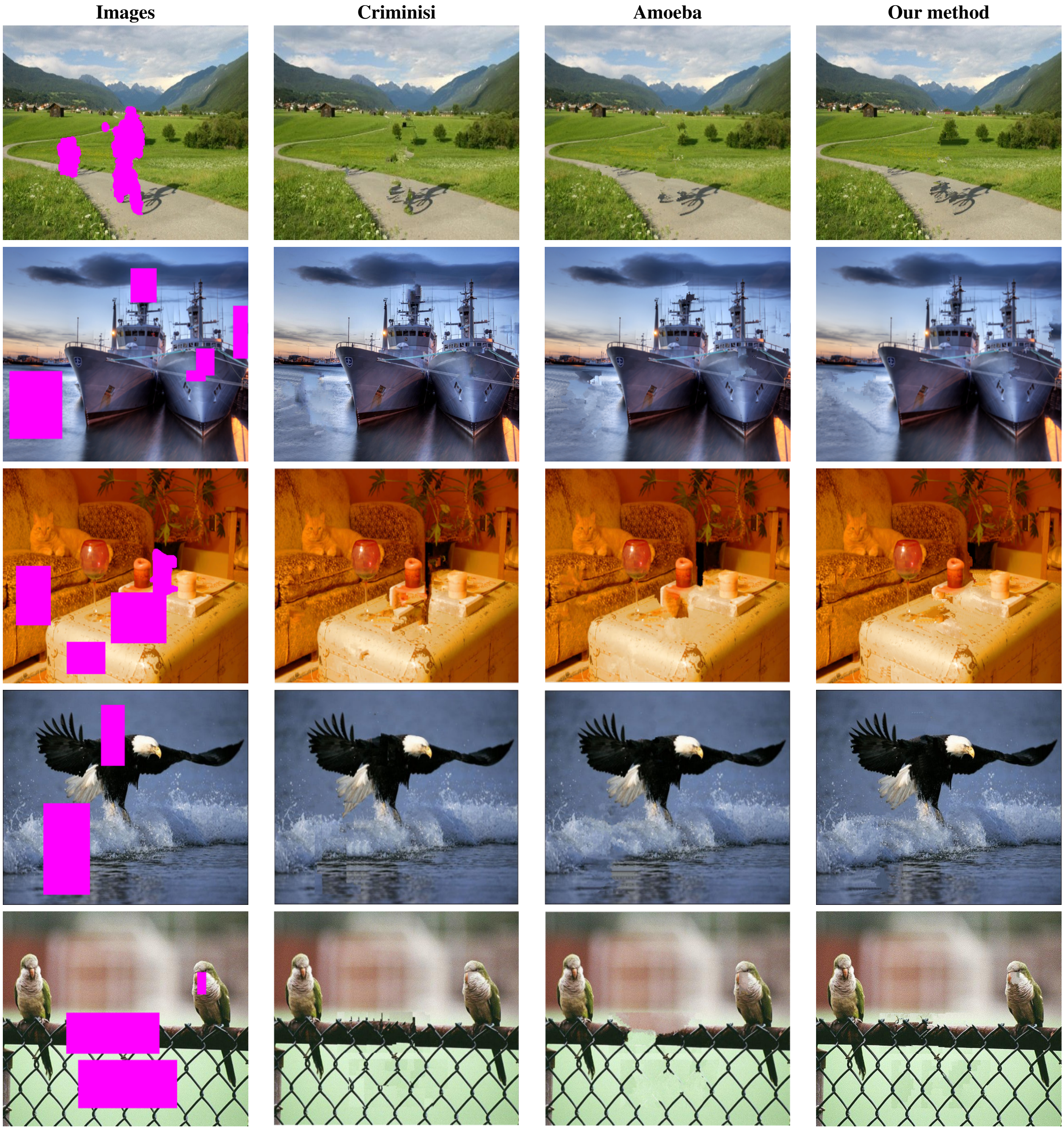


Figure 9: *A subjective selection of the most plausible results for some images*

results have fewer inconsistent artifacts, and the restored images are more plausible.

The innovations proposed in this research (the priority function, the construction of the patch, and the search for the best match) can be used in any other modular version of the exemplar-based inpainting method based on the Criminisi pipeline. Although having a complexity close to that of the basic Criminisi algorithm, a naïve implementation of our method can quickly increase the time of the restoration considerably.

Acknowledgments

The authors would like to thank all the TikiLab members for their support.

_

References

[ANC12] AL-NAJJAR Y., CHEN S. D.: Comparison of image quality assessment: Psnr, hvs, ssim, uiqi. *International Journal of Scientific & Engineering Research 3* (01 2012), 1–5. 7

[BBCS10] BUGEAU A., BERTALMÍO M., CASELLES V., SAPIRO G.: A comprehensive framework for image inpainting. *IEEE Transactions on Image Processing 19*, 10 (2010), 2634–2645. doi:10.1109/TIP.2010.2049240. 3

[BBS01] BERTALMIO M., BERTOZZI A. L., SAPIRO G.: Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (2001), vol. 1, pp. I–I. doi:10.1109/CVPR.2001.990497. 1, 2

[BDTL15] BUYSSENS P., DAISY M., TSCHUMPERLE D., LEZORAY O.: Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society 24* (03 2015). doi:10.1109/TIP.2015.2411437. 2

[BEG07] BERTOZZI A. L., ESEDOGLU S., GILLETTE A.: Inpainting of binary images using the cahn–hilliard equation. *IEEE Transactions on Image Processing 16*, 1 (2007), 285–291. doi:10.1109/TIP.2006.887728. 2

[BHS09] BURGER M., HE L., SCHÖNLIEB C.-B.: Cahn-hilliard inpainting and a generalization for grayvalue images. *SIAM J. Imaging Sciences 2* (01 2009), 1129–1167. doi:10.1137/080728548. 2

[BSCB00] BERTALMÍO M., SAPIRO G., CASELLES V., BALLESTER C.: Image inpainting. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics* (01 2000), pp. 417–424. 2

[BVSO03] BERTALMIO M., VESE L., SAPIRO G., OSHER S.: Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing 12*, 8 (2003), 882–889. doi:10.1109/TIP.2003.815261. 3

[CCWB18] CASTILLO S., CUNNINGHAM D., WINGER C., BREUSS M.: Morphological amoeba-based patches for exemplar-based inpainting. *Journal of WSCG 26* (06 2018). doi:10.24132/JWSCG.2018.26.2.6. 2, 6

[CDNL98] CHAMBOLLE A., DE VORE R. A., NAM-YONG LEE, LUCIER B. J.: Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing 7*, 3 (1998), 319–335. doi:10.1109/83.661182. 2

[CPT04a] CRIMINISI A., PEREZ P., TOYAMA K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing 13*, 9 (2004), 1200–1212. doi:10.1109/TIP.2004.833105. 1, 2

[CPT04b] CRIMINISI A., PÉREZ P., TOYAMA K.: Region filling and object removal by exemplar-based image inpainting. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society 13* (10 2004), 1200–12. doi:10.1109/TIP.2004.833105. 3, 6

[CRG*13] CORNELIS B., RUŽIĆ T., GEZELS E., DOOMS A., PIŽURICA A., PLATIŠA L., CORNELIS J., MARTENS M., DE MEY M., DAUBECHIES I.: Crack detection and inpainting for virtual restoration of paintings: The case of the ghent altarpiece. *Signal Processing 93*, 3 (2013), 605–619. Image Processing for Digital Art Work. URL: https://www.sciencedirect.com/science/article/pii/S0165168412002526, doi:https://doi.org/10.1016/j.sigpro.2012.07.022. 2

[CS01] CHAN T., SHEN J.: Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation 12* (12 2001), 436–449. doi:10.1006/jvci.2001.0487. 2

[CSL*17] CAI N., SU Z., LIN Z., WANG H., YANG Z., LING B.: Blind inpainting using the fully convolutional neural network. *The Visual Computer 33* (02 2017). doi:10.1007/s00371-015-1190-z. 3

[DJL*12] DONG B., JI H., LI J., SHEN Z., XU Y.: Wavelet frame based blind image inpainting. *Applied and Computational Harmonic Analysis 32*, 2 (2012), 268–279. URL: https://www.sciencedirect.com/science/article/pii/S1063520311000765, doi:https://doi.org/10.1016/j.acha.2011.06.001. 2

[DSC04] DEMANET L., SONG B., CHAN T.: Image inpainting by correspondence maps: a deterministic approach. *Applied and Computational Mathematics 1100* (07 2004). accessed 03/3/2021. URL: https://math.mit.edu/icg/papers/demanet_song_chan.pdf. 2

[ESQD05] ELAD M., STARCK J.-L., QUERRE P., DONOHO D.: Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis 19*, 3 (2005), 340–358. Computational Harmonic Analysis - Part 1. URL: https://www.sciencedirect.com/science/article/pii/S1063520305000655, doi:https://doi.org/10.1016/j.acha.2005.03.005. 3

[FZ18] FAN Q., ZHANG L.: A novel patch matching algorithm for exemplar-based image inpainting. *Multimedia Tools and Applications 77* (05 2018). doi:10.1007/s11042-017-5077-z. 2

[GG84] GEMAN S., GEMAN D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*, 6 (1984), 721–741. doi:10.1109/TPAMI.1984.4767596. 2

[GGGD14] GOTHWAL R., GUPTA S., GUPTA D., DAHIYA A. K.: Color image segmentation algorithm based on rgb channels. In *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization* (2014), pp. 1–5. doi:10.1109/ICRITO.2014.7014669. 3

[GL14] GUILLEMOT C., LE MEUR O.: Image inpainting : Overview and recent advances. *IEEE Signal Processing Magazine 31*, 1 (2014), 127–144. doi:10.1109/MSP.2013.2273004. 1, 2

[GPAM*14] GOODFELLOW I. J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2014), NIPS'14, MIT Press, p. 2672–2680. 3

[Gro04] GROSSAUER H.: A combined pde and texture synthesis approach to inpainting. In *Proc ECCV* (05 2004), vol. 3022, pp. 214–224. doi:10.1007/978-3-540-24671-8_17. 3

[GS03] GROSSAUER H., SCHERZER O.: Using the complex ginzburg-landau equation for digital inpainting in 2d and 3d. vol. 2695, pp. 225–236. doi:10.1007/3-540-44935-3_16. 3

[ISSI17] IIZUKA S., SIMO-SERRA E., ISHIKAWA H.: Globally and locally consistent image completion. *ACM Trans. Graph. 36*, 4 (July 2017). URL: https://doi.org/10.1145/3072959.3073659, doi:10.1145/3072959.3073659. 3

[JQ08] JIYING WU, QIUQI RUAN: A novel hybrid image inpainting model. In *2008 International Conference on Audio, Language and Image Processing* (2008), pp. 138–142. doi:10.1109/ICALIP.2008.4589952. 3

[KMFR95] KOKARAM A. C., MORRIS R. D., FITZGERALD W. J., RAYNER P. J. W.: Interpolation of missing data in image sequences. *IEEE Transactions on Image Processing 4*, 11 (1995), 1509–1519. doi:10.1109/83.469932. 1

[KSSH14] KÖHLER R., SCHULER C., SCHÖLKOPF B., HARMELING S.: Mask-specific inpainting with deep neural networks. In *Pattern Recognition* (09 2014), pp. 523–534. doi:10.1007/978-3-319-11752-2_43. 3

[KY15] K. Alilou V., Yaghmaee F.: Application of grnn neural network in non-texture image inpainting and restoration. *Pattern Recognition Letters 62* (2015), 24–31. URL: https://www.sciencedirect.com/science/article/pii/S0167865515001476, doi:https://doi.org/10.1016/j.patrec.2015.04.020. 3

[LDM05] Lerallut R., Decencière E., Meyer F.: Image processing using morphological amoebas. *Proceedings of the 5th International Symposium on Mathematical Morphology* (01 2005), 13–22. 4

[LHLC10] Lu Z., Huang H., Li L., Cheng D.: A novel exemplar-based image completion scheme with adaptive tv-constraint. In *2010 Fourth International Conference on Genetic and Evolutionary Computing* (2010), pp. 94–97. doi:10.1109/ICGEC.2010.31. 2

[LMWY13] Liu J., Musialski P., Wonka P., Ye J.: Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 1 (2013), 208–220. doi:10.1109/TPAMI.2012.39. 1

[LZW03] Levin, Zomet, Weiss: Learning how to inpaint from global image statistics. In *Proceedings Ninth IEEE International Conference on Computer Vision* (2003), pp. 305–312 vol.1. doi:10.1109/ICCV.2003.1238360. 2

[MM98] Martens J.-B., Meesters L.: Image dissimilarity. *Signal Processing 70* (11 1998), 155–176. doi:10.1016/S0165-1684(98)00123-6. 6

[MT19] Mousavi P., Tavakoli A.: A new algorithm for image inpainting in fourier transform domain. *Comp. Appl. Math. 38*, 22 (2019). doi:https://doi.org/10.1007/s40314-019-0761-4. 2

[OLKK19] Ouattara N., Loum G., Koffi G., Kodjo A.: A new image inpainting approach based on criminisi algorithm. *International Journal of Advanced Computer Science and Applications 10* (01 2019). doi:10.14569/IJACSA.2019.0100655. 2

[PKD*16] Pathak D., Krähenbühl P., Donahue J., Darrell T., Efros A.: Context encoders: Feature learning by inpainting. doi:arXiv:1604.07379. 2, 3

[PPM12] Patel P., Prajapati A., Mishra S.: Review of different inpainting algorithms. *International Journal of Computer Applications 59* (12 2012), 30–34. doi:10.5120/9650-4411. 2

[QHX17] Qiang Z., He L., Xu D.: Exemplar-based pixel by pixel inpainting based on patch shift. In *Computer Vision* (11 2017), pp. 370–382. doi:10.1007/978-981-10-7302-1_31. 2

[SC02] Shen J., Chan T.: Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics 62* (10 2002), 1019–1043. doi:10.1137/S0036139900368844. 2

[Sch15] Schönlieb C.-B.: *Partial Differential Equation Methods for Image Inpainting*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2015. doi:10.1017/CBO9780511734304. 2

[SWFY20] Shao H., Wang Y., Fu Y., Yin Z.: Generative image inpainting via edge structure and color aware fusion. *Signal Processing: Image Communication 87* (2020), 115929. URL: https://www.sciencedirect.com/science/article/pii/S0923596520301193, doi:https://doi.org/10.1016/j.image.2020.115929. 2

[Tel04] Telea A.: An image inpainting technique based on the fast marching method. *Journal of Graphics Tools 9*, 1 (2004), 23–34. URL: https://doi.org/10.1080/10867651.2004.10487596, doi:10.1080/10867651.2004.10487596. 2

[VSB19] Vitoria. P., Sintes. J., Ballester. C.: Semantic image inpainting through improved wasserstein generative adversarial networks. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,* (2019), INSTICC, SciTePress, pp. 249–260. doi:10.5220/0007367902490260. 3

[WB02] Wang Z., Bovik A.: A universal image quality index. *Signal Processing Letters, IEEE 9* (04 2002), 81 – 84. doi:10.1109/97.995823. 6, 7

[WL00a] Wei L.-Y., Levoy M.: Fast texture synthesis using tree-structured vector quantization. *Computer Graphics (Proceedings of SIGGRAPH'00) 34* (05 2000). doi:10.1145/344779.345009. 2

[WL00b] Wei L.-Y., Levoy M.: Fast texture synthesis using tree-structured vector quantization. *Computer Graphics (Proceedings of SIGGRAPH'00) 34* (05 2000). doi:10.1145/344779.345009. 3

[WPP02] Won C. S., Park D. K., Park S.-J.: Efficient use of mpeg-7 edge histogram descriptor. *ETRI Journal 24*, 1 (2002), 23–30. URL: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.02.0102.0103, doi:https://doi.org/10.4218/etrij.02.0102.0103. 6, 7

[XS10] Xu Z., Sun J.: Image inpainting by patch propagation using patch sparsity. *IEEE Transactions on Image Processing 19*, 5 (2010), 1153–1165. doi:10.1109/TIP.2010.2042098. 2

[YLL*18] Yan Z., Li X., Li M., Zuo W., Shan S.: Shift-net: Image inpainting via deep feature rearrangement. *Computer Vision and Pattern Recognition* (01 2018). URL: https://arxiv.org/abs/1801.09392. 3

[YLY*18] Yu J., Lin Z., Yang J., Shen X., Lu X., Huang T. S.: Generative image inpainting with contextual attention. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 5505–5514. doi:10.1109/CVPR.2018.00577. 3